Structural and Multidisciplinary Optimization (2025) 68:188
https://doi.org/10.1007/s00158-025-04129-1

RESEARCH PAPER

Topology optimization for multi-component robotic arms under time-varying loads

Chi Wu[1,2] · Yanan Xu[2] · Jianguang Fang[3] · Guangyong Sun[4] · Grant P. Steven[2] · Qing Li[2]

Received: 20 February 2025 / Revised: 29 July 2025 / Accepted: 21 August 2025 / Published online: 13 September 2025

Abstract
High structural performance of robotic arms is essential for positioning precision and energy efficiency in industrial applications. This study presents a design-dependent topology optimization framework for multi-component robotic arms, addressing dynamic multi-load scenarios during motion. Utilizing the Bi-directional Evolutionary Structural Optimization (BESO) method, the framework incorporates self-weight and inertial loads and takes into account the time-varying acceleration of serial robotic arms in a forward manner. The objective is to minimize mean compliance under a transient context. Three numerical examples, namely a 2D single robotic arm, a 3D two-arm in-plane rotation, and a 3D three-arm spatial movement, are presented to demonstrate the efficacy of the proposed framework. The optimization results highlight the necessity of considering design-dependent loads and different robotic configurations in topological evolution. Comparative studies on motion strategies and travel times further emphasize the importance of integrating structural performance with movement dynamics. This approach offers significant insights into the topological design of robotic arms, potentially improving their operational efficiency and precision in various industrial applications.

Keywords Topology optimization · Robotic arms · Design-dependent · Time-varying · Bi-directional evolutionary structural optimization (BESO)

1 Introduction

Industrial robots are widely utilized across various fields to enhance production efficiency and reduce labor cost, particularly involving high precision and repetition, such as manufacturing and assembly in automotive industry (Křepelka et al. 2024), harvesting and picking in agriculture (Yoshida et al. 2022), civil infrastructure construction (Shi et al. 2023), and biomedical surgery (Dagnino and Kundrat 2024). The most common commercial robots are equipped with arm-like structures that can be programmed to mimic human arm functions for a variety of tasks (Lu et al. 2012). There is a fast-growing demand for high-performance robotic arms in these areas, which includes lightweight designs to reduce power consumption (Lee and Seo 2017; Yin et al. 2019; Zhong et al. 2023), increase in high operational speed for efficiency (Alkhafaji 2021), heighten accuracy for target positions (Lattanzi et al. 2020), and a desirable payload-to-weight ratio (Liu and Paulino 2019).

The flexibility resulting from lightweight robotic arms can however lead to escalated oscillatory behavior at their end-effectors, making precise pointing or tip positioning challenging. To address this issue, advanced control techniques such as feedforward and/or feedback control strategies are commonly employed to compensate for dynamic errors arising from both the servo system and the arms. Another proper strategy for reducing residual vibration of end-effectors is to

Responsible Editor: Jianbin Du.

✉ Chi Wu
chi.wu@newcastle.edu.au

✉ Qing Li
qing.li@sydney.edu.au

[1] School of Engineering, University of Newcastle, Callaghan, NSW, Australia

[2] School of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney, Sydney, NSW 2006, Australia

[3] School of Civil and Environmental Engineering, University of Technology Sydney, Sydney, NSW 2007, Australia

[4] State Key Laboratory of Advanced Design and Manufacture for Vehicle Body, Hunan University, Changsha 410082, China

perform structural optimization of robotic arms, especially those used in high-speed motion applications. While recent studies have primarily focused on development of various advanced control systems for robotic arms (Lou et al. 2024; Vicentini 2021), there has been limited research on optimization of arm structures for improving positioning precision subject to high-speed motion. For example, Canh et al. (2024) performed shape optimization for a dual-arm robot, in which the objective was to minimize weight subject to displacement and stress constraints under linear static analysis. In addition, Kuruvilla et al. (2024) conducted a trial-and-error design to minimize the deflection of the end-effector of a robotic arm subject to static analysis. Yue et al. (2024) proposed a design of robotic arms composed of tensegrity-inspired modules capable of achieving desired deformations. All these studies highlighted the growing demand for efficient design optimization approaches tailored for robotic arms with improved lightweight performance and accuracy.

In the past three decades, topology optimization has been extensively developed and applied across various engineering disciplines (Wu et al. 2017, 2019, 2020; Xie et al. 2023). Typical topology optimization methods can be classified into two groups, namely density-based and boundary-based algorithms. The density-based approaches primarily include the Solid Isotropic Material with Penalization (SIMP) method (Bendsøe and Sigmund 1999) and its extended forms, as well as the Evolutionary Structural Optimization (ESO) approach (Xie and Steven 1993) and its later Bi-directional version (BESO) (Huang and Xie 2010a, b). On the other hand, boundary-based approaches encompass the level-set method (Allaire et al. 2004; Wang et al. 2003), the moving morphable component approach (Zhang et al. 2016). In literature, topology optimization has been employed in design of robotic arms. For instance, Kishida and Kurahashi (2023) employed the SIMP method to optimize a single robotic arm to minimize its mean compliance and maximum von Mises stress subject to a volume constraint. Deng et al. (2018) applied the BESO method to the lightweight design of a six-degree-of-freedom robotic arm, aiming to maximize the stiffness of the arm while adhering to volume and natural frequency constraints. Furthermore, Alkalla and Fanni (2021) discovered that the mass moment of inertia of robotic arms significantly affects preset time-optimal control strategies. Consequently, they investigated the topology optimization of robotic arms to minimize traveling time while maintaining a certain deflection constraint at the end point of the arm. Other studies have introduced topology optimization for the lightweight design of robotic arms in a linear static context (Paska et al. 2020; Srinivas and Javed 2020a, b).

In practice, the kinematic configurations of robotic arms change over time, especially for those with multiple linked arms whose relative spatial positions change significantly during motion (Bowling and Kim 2006). Therefore, topology optimization with only a single load case may be insufficient to ensure the desired functional performance of a robotic arm throughout its entire operational process. To address this issue, one approach is to identify an extreme load or worst-case scenario during the motion of robotic arms. For instance, Chong et al. (2020) proposed to identify the worst-case scenario of a robotic arm, followed by topology optimization using the SIMP method to minimize mean compliance in the worst case. Similarly, Liu et al. (2022) performed topology optimization for the lightweight design of robotic arms subjected to an identified extreme load case. However, robotic arms designed for worst-case conditions may still be ineffective for other angular/linear positions. Another approach is to consider multiple load cases extracted at different time points in the topology optimization process. For example, Srinivas and Javed (2020a, b) conducted topology optimization for robotic arms by considering multiple configurations with various load cases. They adopted a superposition method to combine the generated topologies from different configurations into a single optimal topology. Additionally, they investigated SIMP-based topology optimization to maximize the stiffness of robotic arms subject to diverse dynamic loads using the equivalent static load method (Srinivas and Javed 2021).

Self-weight and inertial loads can significantly affect the functional performance of robots that typically require high precision and high efficiency, particularly for load-carrying robots with high-speed motion, which can play a crucial role in dynamic deflection of end-effectors. Despite their importance, unfortunately, few studies have adopted these time-varying and design-dependent loads in topology optimization for robotic arms. To tackle this issue, this study develops a transient topology optimization framework for robotic arms that accounts for time-varying inertial and self-weight loads. The BESO method is employed to maximize the mean compliance of multi-component robotic arms subject to a volume fraction constraint for energy efficiency. To calculate the inertial loads of multi-component robotic arms, the time-varying acceleration of mass points in each arm is determined in a stepwise fashion. A design-dependent sensitivity analysis is developed to account for inertial and self-weight loads. In this paper, both 2D and 3D illustrative examples are provided to demonstrate the effectiveness and key features of the proposed topology optimization strategy. To highlight the importance of considering time-varying and design-dependent loads, we compare the design results obtained from the proposed approach with those generated from conventional topology optimization without considering inertial and self-weight effects. Additionally, we explore the effects of various motions and travel times on the generated topologies to provide further insights into such a transient optimization framework.

The remainder of the paper is organized as follows: Sect. 2 presents the BESO method and the numerical modeling for considering inertial and self-weight loads, including the stepwise calculation of acceleration for each mass point. Section 3 presents the 2D and 3D numerical examples for the design of robotic arms, along with discussions, followed by conclusions drawn in Sect. 4.

2 Methodology

2.1 The BESO method and material interpolation scheme

The BESO method is an extended version of the original Evolutionary Structural Optimization (ESO) method proposed by Xie and Steven (1993) in the 1990s. As a density algorithm, each element e in the design domain is associated with a design variable x_e, where $x_e = 1$ denotes a solid element and $x_e = x_{\min}$ represents a low-stiffness, soft element with a small value of $x_{\min}$ to prevent numerical singularity during finite element (FE) analyses.

The conventional BESO method can adopt a power-law based material interpolation scheme (Huang and Xie 2010a, b), which is also commonly used in the SIMP method (Bendsøe and Sigmund 1999). The basic idea of the BESO method is to gradually delete and/or add elements according to their sensitivities to the design criteria until convergence is achieved for desired objective and constraint functions. In contrast, the SIMP method uses continuous density variables ranging from 0 to 1 and typically employs optimality criteria or gradient-based algorithms to iteratively update these variables.

However, previous studies have found that the power-law model may lead to unbounded material properties in low-density regions (e.g., $x_e = x_{\min}$) when considering design-dependent loads (Huang and Xie 2011). This makes it difficult to obtain a clear topological configuration with design variables either 1 or $x_{\min}$ (Bruyneel and Duysinx 2005). Therefore, the Rational Approximation of Material Properties (RAMP) model (Stolpe and Svanberg 2001) is adopted herein to effectively address this issue, as demonstrated in the literature (Gao and Zhang 2010; Wu et al. 2019; Xu et al. 2013). In the RAMP model, the Young's modulus E_e of element e is expressed as

$$E_e = \frac{x_e}{1 + q\left(1 - x_e\right)} E, \tag{1}$$

where E denotes the Young's modulus of solid materials, and q is a penalization factor associated with the RAMP model.

To account for inertial and self-weight loads, the material density ρ_e of element e is interpolated linearly as

$$\rho_e = x_e \rho, \tag{2}$$

where ρ is the material density of solid material.

2.2 Finite element analysis for design-dependent loads

As robotic arms assume different angular positions while moving from the start to end time points, we divide the entire process into several time points and treat the kinematic configuration at each time point with specific inertial and self-weight loads. The idea is similar to that of the equivalent static load (ESL) method accounting for inertial and dynamic forces (Park 2011). Nevertheless, we directly incorporate the inertial load as a distributed body force derived from the local acceleration and material density. At each time step, the elemental stiffness matrix $\mathbf{k}_e$, based on the current geometry remains unchanged, is expressed as

$$\mathbf{k}_e = \int_{\Omega_e} \mathbf{B}_e^{\mathrm{T}} \mathbf{C}_e \mathbf{B}_e \mathrm{d}\Omega_e, \tag{3}$$

where Ω_e denotes the volume of element e. $\mathbf{B}_e$ is the strain–displacement matrix, which varies according to the type of elements (Reddy 1993). $\mathbf{C}_e$ is the material constitutive matrix of element e, which is assumed to be isotropic material properties in this study. It can be calculated using Young's modulus E_e and Poisson's ν. For a 3D case, one can have

$$\mathbf{C}_e = \frac{x_e}{1 + q\left(1 - x_e\right)} \mathbf{C}, \tag{4}$$

$$\mathbf{C} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & 1-2\nu & 0 & 0 \\ 0 & 0 & 0 & 0 & 1-2\nu & 0 \\ 0 & 0 & 0 & 0 & 0 & 1-2\nu \end{bmatrix}, \tag{5}$$

where ν denotes Poisson's ratio of solid materials and $\mathbf{C}$ denotes the material constitutive matrix for solid elements. The elemental stiffness matrix $\mathbf{k}_e$ can be further expressed as

$$\mathbf{k}_e = \frac{x_e}{1 + q\left(1 - x_e\right)} \int_{\Omega_e} \mathbf{B}_e^{\mathrm{T}} \mathbf{C} \mathbf{B}_e \mathrm{d}\Omega_e = \frac{x_e}{1 + q\left(1 - x_e\right)} \mathbf{k}_e^0, \tag{6}$$

where $\mathbf{k}_e^0$ is the elemental stiffness matrix of element e of solid materials.

The self-weight load of element e generated by the gravity can be calculated as

$$\mathbf{f}_e^g = \int_{\Omega_e} \rho_e \mathbf{N}_e^{\mathrm{T}} \mathbf{g} \mathrm{d}\Omega_e, \tag{7}$$

where $\mathbf{f}_e^g$ and $\mathbf{N}_e$ are the nodal force vector caused by gravity and the matrix of shape functions of element e, respectively. $\mathbf{g}$ denotes the vector of gravitational acceleration, given as

$$\mathbf{g} = \begin{cases} \begin{bmatrix} 0 & g \end{bmatrix}^{\mathrm{T}}, & \text{for 2D cases} \\ \begin{bmatrix} 0 & 0 & g \end{bmatrix}^{\mathrm{T}}, & \text{for 3D cases} \end{cases}, \tag{8}$$

where g is the scalar constant of gravity.

Eq. (7) can be further rewritten by substituting Eq. (2), expressed as

$$\mathbf{f}_e^g = x_e \int_{\Omega_e} \rho \mathbf{N}_e^{\mathrm{T}} \mathbf{g} \mathrm{d}\Omega_e. \tag{9}$$

Similarly, the inertial load due to the kinematic acceleration of robotic motion can be calculated as

$$\mathbf{f}_e^\alpha = x_e \int_{\Omega_e} \rho \mathbf{N}_e^{\mathrm{T}} \ddot{\boldsymbol{\alpha}}_e \mathrm{d}\Omega_e, \tag{10}$$

where $\mathbf{f}_e^\alpha$ is the nodal force caused by the kinematic acceleration, and $\ddot{\boldsymbol{\alpha}}_e$ denotes the linear acceleration of element e. $\ddot{\boldsymbol{\alpha}}_e$ can be calculated in a stepwise fashion to be detailed in Sect. 2.3.

In a FE framework, the global equilibrium equation of the robotic system can be expressed as

$$\mathbf{KU} = \mathbf{F}, \tag{11}$$

where $\mathbf{U}$ and $\mathbf{F}$ denote the global nodal displacement and load vector, respectively. $\mathbf{K}$ is the global stiffness matrix, assembled as

$$\mathbf{K} = \sum_{e=1}^{N} \mathbf{k}_e, \tag{12}$$

where N denotes the total number of elements in the system.

As in Eq. (11), $\mathbf{F}$ represents the global force vector that is composed of external force vector and internal body force parts, calculated as

$$\mathbf{F} = \sum_{e=1}^{N} \mathbf{f}_e^\alpha + \sum_{e=1}^{N} \mathbf{f}_e^g + \mathbf{F}^{\mathrm{o}}, \tag{13}$$

where $\mathbf{F}^{\mathrm{o}}$ is the global external force vector that is design independent. $\sum_{e=1}^{N}(\mathbf{f}_e^\alpha + \mathbf{f}_e^g)$ is the design-dependent load.

2.3 Acceleration of robotic arms

In this study, we focus on the most common industrial robots, which are designed as a series of links connected by joints. The velocity and acceleration of each mass point in these links can be calculated in a nonlinear transient manner from the base joint to the end. The base joint here denotes the first joint that connects the link to a fixed surface or base. Without loss of generality, we take Link i and Link $(i+1)$ as an illustrative example to detail the method for calculating the acceleration of serial robots. Figure 1 depicts a serial robot with two links connected by joints. We take point $\mathbf{P}_{i+1}$ within Link $(i+1)$ to derive its acceleration.

Let $\mathbf{w}_i$ and $\mathbf{w}_{i+1}$ represent the angular velocities of Link i and Link $(i+1)$ with respect to the global coordinate system defined by the x, y and z axes, as shown in Fig. 1. The angular velocity $\mathbf{w}_{i+1}$ can be calculated from $\mathbf{w}_i$ as follows:

$$\mathbf{w}_{i+1} = \mathbf{w}_i + \dot{\theta}_{j+1}\mathbf{Z}_{j+1}, \tag{14}$$

where $\mathbf{Z}_j$ and $\mathbf{Z}_{j+1}$ are the unit vectors of the axes of rotation for Joint j and Joint $(j+1)$, respectively. $\dot{\theta}_j$ and $\dot{\theta}_{j+1}$ are the angular velocities for Joint j and Joint $(j+1)$ along directions $\mathbf{Z}_j$ and $\mathbf{Z}_{j+1}$, respectively.

Let $\mathbf{O}_j$ and $\mathbf{O}_{j+1}$ denote the centers of rotation for Joint j and Joint $(j+1)$, respectively. The linear velocity $\dot{\mathbf{O}}_{j+1}$ at $\mathbf{O}_{j+1}$ can be calculated as

$$\dot{\mathbf{O}}_{j+1} = \dot{\mathbf{O}}_j + \mathbf{w}_i \times \mathbf{r}_{O_j,O_{j+1}}, \tag{15}$$

where $\dot{\mathbf{O}}_j$ is the linear velocity at $\mathbf{O}_j$ and $\mathbf{r}_{O_j,O_{j+1}}$ denotes the vector from point $\mathbf{O}_j$ to point $\mathbf{O}_{j+1}$. Therefore, the angular and linear velocities at point $\mathbf{P}_{i+1}$ within Link $(i+1)$ can be calculated as

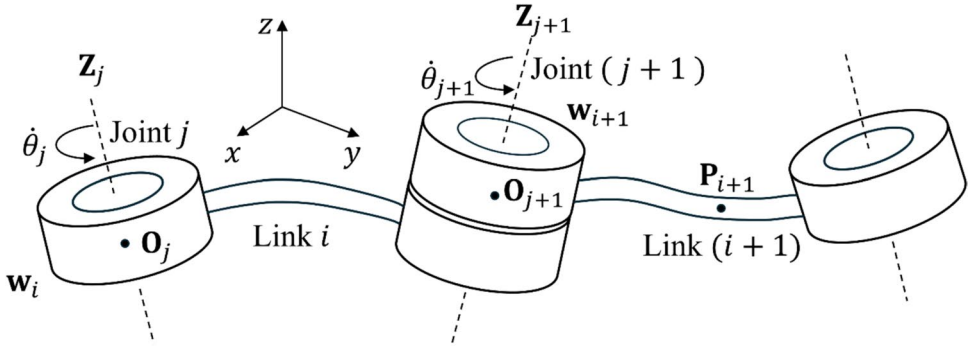

Fig. 1 A schematic diagram of a serial robot with two links $(i, i+1)$ and two joints $(j, j+1)$

$$\mathbf{w}_{P_{i+1}} = \mathbf{w}_{i+1}, \tag{16}$$

$$\dot{\boldsymbol{\alpha}}_{P_{i+1}} = \dot{\mathbf{O}}_{j+1} + \mathbf{w}_{i+1} \times \mathbf{r}_{O_{j+1},P_{i+1}} \tag{17}$$

where $\mathbf{w}_{P_{i+1}}$ and $\dot{\boldsymbol{\alpha}}_{P_{i+1}}$ are the angular and linear velocities at point $\mathbf{P}_{i+1}$, and $\mathbf{r}_{O_{j+1},P_{i+1}}$ represents the vector from point $\mathbf{O}_{j+1}$ to an arbitrary point $\mathbf{P}_{i+1}$.

To determine the angular acceleration $\dot{\mathbf{w}}_{i+1}$ of Link $(i+1)$, we differentiate Eq. (14) with respect to time, resulting in

$$\dot{\mathbf{w}}_{i+1} = \dot{\mathbf{w}}_i + \ddot{\theta}_{j+1}\mathbf{Z}_{j+1} + \dot{\theta}_{j+1}\dot{\mathbf{Z}}_{j+1}, \tag{18}$$

where $\dot{\mathbf{w}}_i$ is the angular acceleration of Link i, and $\ddot{\theta}_{j+1}$ represents the angular acceleration scalar for Joint $(j+1)$ with respect to $\mathbf{Z}_{j+1}$. $\dot{\mathbf{Z}}_{j+1}$ indicates the spatial change of $\mathbf{Z}_{j+1}$ with respect to time, which can be calculated as

$$\dot{\mathbf{Z}}_{j+1} = \mathbf{w}_i \times \mathbf{Z}_{j+1}, \tag{19}$$

Substituting Eq. (19) into Eq. (18), we obtain

$$\dot{\mathbf{w}}_{i+1} = \dot{\mathbf{w}}_i + \ddot{\theta}_{j+i}\mathbf{Z}_{j+1} + \dot{\theta}_{j+i}\mathbf{w}_i \times \mathbf{Z}_{j+1}, \tag{20}$$

For the linear acceleration at point $\mathbf{O}_{j+1}$, we can differentiate Eq. (15) with respect to time, calculated as

$$\ddot{\mathbf{O}}_{j+1} = \ddot{\mathbf{O}}_j + \dot{\mathbf{w}}_i \times \mathbf{r}_{O_j,O_{j+1}} + \mathbf{w}_i \times \dot{\mathbf{r}}_{O_j,O_{j+1}}, \tag{21}$$

where $\ddot{\mathbf{O}}_{j+1}$ and $\ddot{\mathbf{O}}_j$ are the linear acceleration at point $\mathbf{O}_j$ and $\mathbf{O}_{j+1}$, respectively. $\dot{\mathbf{r}}_{O_j,O_{j+1}}$ denotes the vector from point $\mathbf{O}_j$ to $\mathbf{O}_{j+1}$, which equals to

$$\dot{\mathbf{r}}_{O_j,O_{j+1}} = \dot{\mathbf{O}}_{j+1} - \dot{\mathbf{O}}_j. \tag{22}$$

According to Eq. (15), Eq. (22) can be rewritten as

$$\dot{\mathbf{r}}_{O_j,O_{j+1}} = \mathbf{w}_i \times \mathbf{r}_{O_j,O_{j+1}}. \tag{23}$$

Substitution of Eq. (23) into Eq. (21) yields

$$\ddot{\mathbf{O}}_{j+1} = \ddot{\mathbf{O}}_j + \dot{\mathbf{w}}_i \times \mathbf{r}_{O_j,O_{j+1}} + \mathbf{w}_i \times \left(\mathbf{w}_i \times \mathbf{r}_{O_j,O_{j+1}}\right), \tag{24}$$

The linear acceleration of point $\mathbf{P}_{i+1}$ can then be calculated in a similar manner, expressed as

$$\ddot{\boldsymbol{\alpha}}_{P_{i+1}} = \ddot{\mathbf{O}}_{j+1} + \dot{\mathbf{w}}_{i+1} \times \mathbf{r}_{O_{j+1},P_{i+1}} + \mathbf{w}_{i+1} \times \left(\mathbf{w}_{i+1} \times \mathbf{r}_{O_{j+1},P_{i+1}}\right). \tag{25}$$

It can be noted that $\ddot{\boldsymbol{\alpha}}_{P_{i+1}}$ can be calculated in a forward fashion from the base joint (the first joint fixed to ground surface) to Link i and Link $(i+1)$. Subsequently, $\ddot{\boldsymbol{\alpha}}_e$ can be determined with respect to each link in a serial robot to account for the inertial load as defined in Eq. (10).

2.4 Optimization problem

This study aims to optimize the topology of arms for serial robots with I links. To account for the overall travel time under various configurations and loading conditions, the entire motion of the serial robot arm is divided into different time steps t. The objective is to minimize the mean compliance of the robotic arms at each time point t subject to a volume constraint applied to each Link i. This aims to maximize the overall stiffness of the robotic arms throughout the entire motion while maintaining lightweight performance. The optimization problem can be mathematically expressed as follows:

$$\begin{cases} \text{Minimize } C^{\chi} = \sum_{t=1}^{T} \frac{\frac{1}{2}(\mathbf{F}^{(t)})^{\mathrm{T}}\mathbf{U}^{(t)}}{C_{initial}^{(t)}} \\ \text{Subject to } \mathbf{K}^{(t)}\mathbf{U}^{(t)} = \mathbf{F}^{(t)} \\ \frac{\sum_{e=1}^{N_i} V_e^{(i)} x_e^{(i)}}{V^{(i)*}} - V_f^{(i)} = 0 \\ x_e^{(i)} = x_{\min} \text{ or } 1 \end{cases}, \tag{26}$$

where $\mathbf{K}^{(t)}$, $\mathbf{U}^{(t)}$, and $\mathbf{F}^{(t)}$ are the global stiffness matrix, global displacement vector, and global force vector at time point t, respectively, as illustrated in Sect. 2.2. $x_e^{(i)}$, $V_e^{(i)}$, and N_i denote the design variable, the volume of element e, and the total number of elements in the design domain of Link i. $V^{(i)*}$ is the initial total volume for Link i when all $x_e^{(i)}$ are equal to 1. $V_f^{(i)}$ is the prescribed volume fraction for Link i. C^{χ} denotes the total mean compliance, calculated by sum of all the mean compliance at each time point t and normalized by $C_{initial}^{(t)}$. The normalized factors $C_{initial}^{(t)}$ are chosen as the initial mean compliance at each time point t when optimization starts. In this study, discrete time points t are selected to represent distinct robotic configurations during motion without modeling the full structural dynamic behavior over time. As such, transient effects are not directly considered in the optimization problem. It is noted that the full transient analysis is expected to be further studied for the time-dependent optimization problem. The full structural dynamic behavior with respect to the time is not directly considered. Such a transient problem needs to be further investigated.

2.5 Sensitivity analysis

The sensitivity number is a key component in the BESO method, which is used to rank and evolve elements by adding or deleting them.

Let one consider a single arbitrary time step t in the objective mean compliance defined in Eq. (26), which can be rewritten as

$$\frac{\frac{1}{2}\left(\mathbf{F}^{(t)}\right)^{\mathrm{T}}\mathbf{U}^{(t)}}{C_{initial}^{(t)}}=\frac{C}{C_{initial}}, \tag{27}$$

where $C=\frac{1}{2}\mathbf{F}^{\mathrm{T}}\mathbf{U}+\boldsymbol{\lambda}^{\mathrm{T}}(\mathbf{F}-\mathbf{K}\mathbf{U})$, $\boldsymbol{\lambda}$ is a Lagrangian multiplier that can be an arbitrary value under the assumption $\mathbf{F}-\mathbf{K}\mathbf{U}=0$. In this section, we use symbols without the superscript $(\cdot)^{(t)}$ to represent a more general time point. The calculation of the sensitivity number at each specific time step follows the procedure detailed in this subsection.

Then, the sensitivity of the modified objective function with respect to design variable x_e is formulated as

$$\frac{\mathrm{d}C}{\mathrm{d}x_e}=\frac{1}{2}\frac{\mathrm{d}\mathbf{F}^{\mathrm{T}}}{\mathrm{d}x_e}\mathbf{U}+\frac{1}{2}\mathbf{F}^{\mathrm{T}}\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}x_e}+\boldsymbol{\lambda}^{\mathrm{T}}\left(\frac{\mathrm{d}\mathbf{F}}{\mathrm{d}x_e}-\frac{\mathrm{d}\mathbf{K}}{\mathrm{d}x_e}\mathbf{U}-\mathbf{K}\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}x_e}\right), \tag{28}$$

where x_e denote $x_e^{(i)}$ in an arbitrary Link i. Equation (28) can be further rewritten by combining the terms $\frac{\mathrm{d}\mathbf{F}^{\mathrm{T}}}{\mathrm{d}x_e}$ and $\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}x_e}$ as

$$\frac{\mathrm{d}C}{\mathrm{d}x_e}=\frac{1}{2}\frac{\mathrm{d}\mathbf{F}^{\mathrm{T}}}{\mathrm{d}x_e}(\mathbf{U}+2\boldsymbol{\lambda})+\left(\frac{1}{2}\mathbf{F}^{\mathrm{T}}-\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{K}\right)\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}x_e}-\boldsymbol{\lambda}^{\mathrm{T}}\frac{\mathrm{d}\mathbf{K}}{\mathrm{d}x_e}\mathbf{U}. \tag{29}$$

To avoid the significant high computational cost of differentiating $\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}x_e}$, the Lagrangian multiplier is chosen as $\boldsymbol{\lambda}=\frac{1}{2}\mathbf{U}$, then Eq. (29) can be simplified as

$$\frac{\mathrm{d}C}{\mathrm{d}x_e}=\frac{\mathrm{d}\mathbf{F}^{\mathrm{T}}}{\mathrm{d}x_e}\mathbf{U}-\frac{1}{2}\mathbf{U}^{\mathrm{T}}\frac{\mathrm{d}\mathbf{K}}{\mathrm{d}x_e}\mathbf{U}, \tag{30}$$

which can be further expressed as

$$\frac{\mathrm{d}C}{\mathrm{d}x_e}=\frac{\mathrm{d}\left(\mathbf{f}_e^{\alpha}+\mathbf{f}_e^{g}\right)^{\mathrm{T}}}{\mathrm{d}x_e}\mathbf{u}_e-\frac{1}{2}\mathbf{u}_e^{\mathrm{T}}\frac{\mathrm{d}\mathbf{k}_e}{\mathrm{d}x_e}\mathbf{u}_e, \tag{31}$$

where $\mathbf{u}_e$ denotes the elemental displacement vector.

According to Eqs. (8) and (10), the first term in Eq. (31) can be formulated as

$$\frac{\mathrm{d}\left(\mathbf{f}_e^{\alpha}+\mathbf{f}_e^{g}\right)^{\mathrm{T}}}{\mathrm{d}x_e}=\int_{\Omega_e}\rho\mathbf{N}_e^{\mathrm{T}}\left(\ddot{\boldsymbol{\alpha}}_e+\mathbf{g}\right)\mathrm{d}\Omega_e. \tag{32}$$

By use of Eqs. (6) and (32), Eq. (30) can be expressed as

$$\frac{\mathrm{d}C}{\mathrm{d}x_e}=\int_{\Omega_e}\rho\mathbf{N}_e^{\mathrm{T}}\left(\ddot{\boldsymbol{\alpha}}_e+\mathbf{g}\right)\mathrm{d}\Omega_e\mathbf{u}_e-\frac{1+q}{2\left[1+q\left(1-x_e\right)\right]^2}\mathbf{u}_e^{\mathrm{T}}\mathbf{k}_e^0\mathbf{u}_e. \tag{33}$$

In the BESO method, the sensitivity number is a key component that measures the relative importance of each element to the objective function and indicates how the objective would be affected if that element were removed or added (Huang and Xie 2008a, b). It can be seen in Eq. (33) that the elemental sensitivity depends on the value of q. In the BESO method, sensitivity numbers denote the relative values of all elemental sensitivities. Therefore, Huang and Xie (2011) suggested defining the sensitivity number of element e by adding the term of $(q+1)$ in the denominator so as to reduce the effects of q, expressed as

$$\psi_e=\frac{1}{q+1}\int_{\Omega_e}\rho\mathbf{N}_e^{\mathrm{T}}\left(\ddot{\boldsymbol{\alpha}}_e+\mathbf{g}\right)\mathrm{d}\Omega_e\mathbf{u}_e-\frac{1}{2\left[1+q\left(1-x_e\right)\right]^2}\mathbf{u}_e^{\mathrm{T}}\mathbf{k}_e^0\mathbf{u}_e. \tag{34}$$

For a solid element with $x_e=1$, ψ_e equals

$$\psi_e=\frac{1}{q+1}\int_{\Omega_e}\rho\mathbf{N}_e^{\mathrm{T}}\left(\ddot{\boldsymbol{\alpha}}_e+\mathbf{g}\right)\mathrm{d}\Omega_e\mathbf{u}_e-\frac{1}{2}\mathbf{u}_e^{\mathrm{T}}\mathbf{k}_e^0\mathbf{u}_e. \tag{35}$$

And the sensitivity number of a soft element with $x_e=x_{\min}$ is calculated as

$$\psi_e=\frac{1}{q+1}\int_{\Omega_e}\rho\mathbf{N}_e^{\mathrm{T}}\left(\ddot{\boldsymbol{\alpha}}_e+\mathbf{g}\right)\mathrm{d}\Omega_e\mathbf{u}_e-\frac{1}{2\left[1+q\left(1-x_{\min}\right)\right]^2}\mathbf{u}_e^{\mathrm{T}}\mathbf{k}_e^0\mathbf{u}_e \tag{36}$$

2.6 Optimization procedure

First, the sensitivity filter scheme is employed to avoid checkerboard patterns (Han et al. 2022), which is expressed as

$$\overline{\psi}_e=\frac{\sum_{m=1}^{M}l\left(\eta_{em}\right)\psi_m}{\sum_{m=1}^{M}l\left(\eta_{em}\right)}, \tag{37}$$

where ψ_m denotes the sensitivity number of element m, whose center is within a filter distance $\eta_{\min}$ to the center of element e. M is the total number of elements within the filter distance $\eta_{\min}$ relative to the element e. $l\left(\eta_{em}\right)$ is a weight factor for element m, defined as

$$l\left(\eta_{em}\right)=\begin{cases}\eta_{\min}-\eta_{em}, & \text{if } \eta_{em}\le\eta_{\min}\\ 0, & \text{if } \eta_{em}>\eta_{\min}\end{cases}, \tag{38}$$

where η_{em} denotes the distance between the center of element e and element m.

To stabilize the optimization process, the sensitivity number of the BESO method is averaged with its historical information after the first iteration (Huang and Xie 2010a, b), defined as

$$\tilde{\psi}_{e,k}=\frac{1}{2}\left(\overline{\psi}_{e,k}+\overline{\psi}_{e,k-1}\right), \tag{39}$$

where k and $(k-1)$ denote the current k^{th} iteration and the previous $(k-1)^{th}$.

Additionally, the targeted volume at each iteration is calculated as

$$V_k=\max\left[V^*,V_{k-1}(1-ER)\right], \tag{40}$$

where V_k and V_{k-1} are the targeted volume at the k^{th} and $(k-1)^{th}$ iterations, respectively. When $k=1, V_0$ denotes the initial volume with all elements equal to 1. ER is the evolutional ratio, representing the deletion ratio before the prescribed volume $V^* = V_f^{(i)} V^{(i)*}$ is achieved. A maximum addition ratio $AR_{\max}$ is also introduced to control the ratio for elements switching from $x_{\min}$ to 1, which can overcome the divergence issue of the BESO method as suggested in Huang and Xie (2008a, b). The convergence criterion of the optimization problem is defined as

$$\frac{\left|\sum_{p=1}^{10}\left(C_{k-p+1}-C_{k-10-p+1}\right)\right|}{\left|\sum_{p=1}^{10} C_{k-p+1}\right|} \le \gamma, \tag{41}$$

where γ is a small tolerance ensuring that the change in the objective function over the last 10 iterations is within an acceptable small value.

The optimization procedure of the BESO method was developed in Querin et al. (1998) and can be briefly outlined as follows:

Step 1: Define the parameters related to the BESO method, including the prescribed volume V^*, evolutionary ratio ER, maximum addition ratio $AR_{\max}$, filter distance $\eta_{\min}$, penalty factor q, and convergence tolerance γ.
Step 2: Perform FE analysis and calculate the sensitivity number.
Step 3: Apply sensitivity filter and average sensitivity using historical information.
Step 4: Calculate the targeted volume V_k for the current k^{th} iteration.
Step 5: Update design variables x_e. For solid elements, x_e is switched from 1 to $x_{\min}$ if $\tilde{\psi}_{e,k} \le \psi_k^{del}$. For soft elements, x_e is switched from $x_{\min}$ to 1 if $\tilde{\psi}_{e,k} > \psi_k^{add}$. Otherwise, x_e remains unchanged. ψ_k^{del} and ψ_k^{add} are threshold parameters which are determined by V_k and $AR_{\max}$.
Step 6: Repeat Steps 2 to 5 until the optimization problem satisfies the convergence criterion defined in Eq. (41).

In Step 5, the threshold parameters ψ_k^{del} and ψ_k^{add} are obtained from the following iterative algorithm (Huang and Xie 2008a, b; Huang et al. 2010; Xia et al. 2018):

1. Let $\psi_k^{del} = \psi_k^{add} = \psi_k^{th}$, where the value of ψ_k^{th} is determined iteratively such that the required target volume V_k is satisfied.
2. Compute the admission ratio AR, which is defined as the volume of elements switching from $x_{\min}$ to 1 divided by the total volume of elements at the current iteration k. If $AR \le AR_{\max}$, skip the next steps; otherwise, ψ_k^{del} and ψ_k^{add} are redetermined in the next steps.
3. Determine ψ_k^{add} iteratively using only the sensitivity numbers of the soft elements until the maximum admission ratio $AR_{\max}$ is met.
4. Determine ψ_k^{del} iteratively using only the sensitivity numbers of the solid elements until the targeted volume V_k is satisfied at the current iteration.

A flowchart on how to implement the proposed BESO approach is also provided in Fig. 2.

3 Results and discussion

In this study, we consider a 2D robotic arm with a single link and two 3D cases of serial robots with two and three links, respectively. The 2D example with one link and the 3D example with two links are assumed to have in-plane rotation, while the 3D example with three links exhibits out-of-plane rotation, aiming to gain a deeper insight. In all numerical examples, the material is assumed to be a mild steel with Young's modulus $E=200$ GPa, Poisson's ratio $\nu = 0.3$, and material density $\rho = 7850\ \mathrm{kg/m^3}$, respectively. The scalar constant of gravitational acceleration g is 9.8 $\mathrm{m/s^2}$. For the parameters used in the BESO procedure, the penalization factor q and convergence tolerance γ are set to be 5 and 0.1%, respectively (Huang and Xie 2011). The

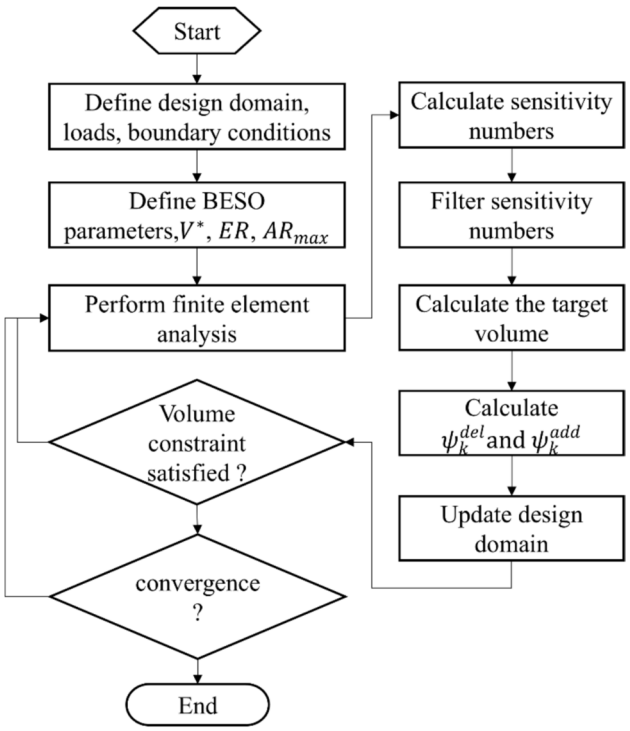

Fig. 2 The flowchart of the BESO procedure

evolutionary ratio $ER = 0.3$ and maximum addition ratio $AR_{\max} = 0.2$ are employed in the 2D case. $ER = 0.2$ and $AR_{\max} = 0.1$ are used in the 3D cases. The filter distances $\eta_{\min}$ are specified for different numerical examples, which are detailed in the following sections. In all examples, objective functions are normalized using the initial mean compliance. The FE analysis is performed using Abaqus (2016), and the sensitivity analysis and BESO optimization procedure are implemented in Matlab (Inc., 2022).

3.1 Example 1: 2D single arm

Example 1 focuses on a 2D robotic arm subject to a plane strain problem. The dimensions of the robotic arm are illustrated in Fig. 3a. The initial arm consists of two holes to represent two revolving joints with rotational centers O_1 and O_2, respectively. The inner radius R_1 is set to be 0.05 m. The outer radius R_3 of the robotic arm is 0.12 m. The distance L between O_1 and O_2 is 0.76 m, and the thickness ξ is 0.2 m. The robotic arm is divided into a design domain (colored in bright cyan) and a non-design domain (colored in dark gray) to maintain the joint features during the optimization. The outer radius R_2 for the non-design domain is set to 0.06 m. The design and non-design domains are meshed with 4-node quadrilateral elements with an average size of 0.004 m (averaging the lengths of all sides of all elements), resulting in a total of 12,436 elements in the design domain and 472 elements in the non-design domain.

The robotic arm is assumed to rotate from the initial position (0° with respect to the x axis) to the end position (90° with respect to the x axis) around center O_1. The motion of robotic arm is provided in Supplementary Movie 1. Figure 3 also depicts the end position of the robotic arm (colored in green). The x and y degrees of freedom (DOFs) of center O_1 are fixed. A concentrated mass point of 100 kg is assumed to be attached to center O_2 to simulate the payload that the robot carries. The angular acceleration $\ddot{\theta}$ and angular velocity $\dot{\theta}$ of the robotic arm are assumed to be constant and linear, as illustrated in Fig. 3b, where $\dot{\theta} = \pi t$ rad/s and $\ddot{\theta} = \pi$ rad/s^2 with a traveling time $T = 1$ s from the initial position to the end position. Regarding the topology optimization problem, the filter radius η_{min} is set to 0.008 m, equal to twice the mesh size. The desired volume $V^{(1)*}$ is assumed to be 40% of the full design domain.

In this example, five different design cases are conducted to comprehensively compare the effects of design-dependent loads and different configurations of the robotic arm on the final topologies. First, a conventional topology optimization is performed for the robotic arm, considering only the initial time point without any design-dependent load. Figure 4 illustrates the iteration history of the objective function and

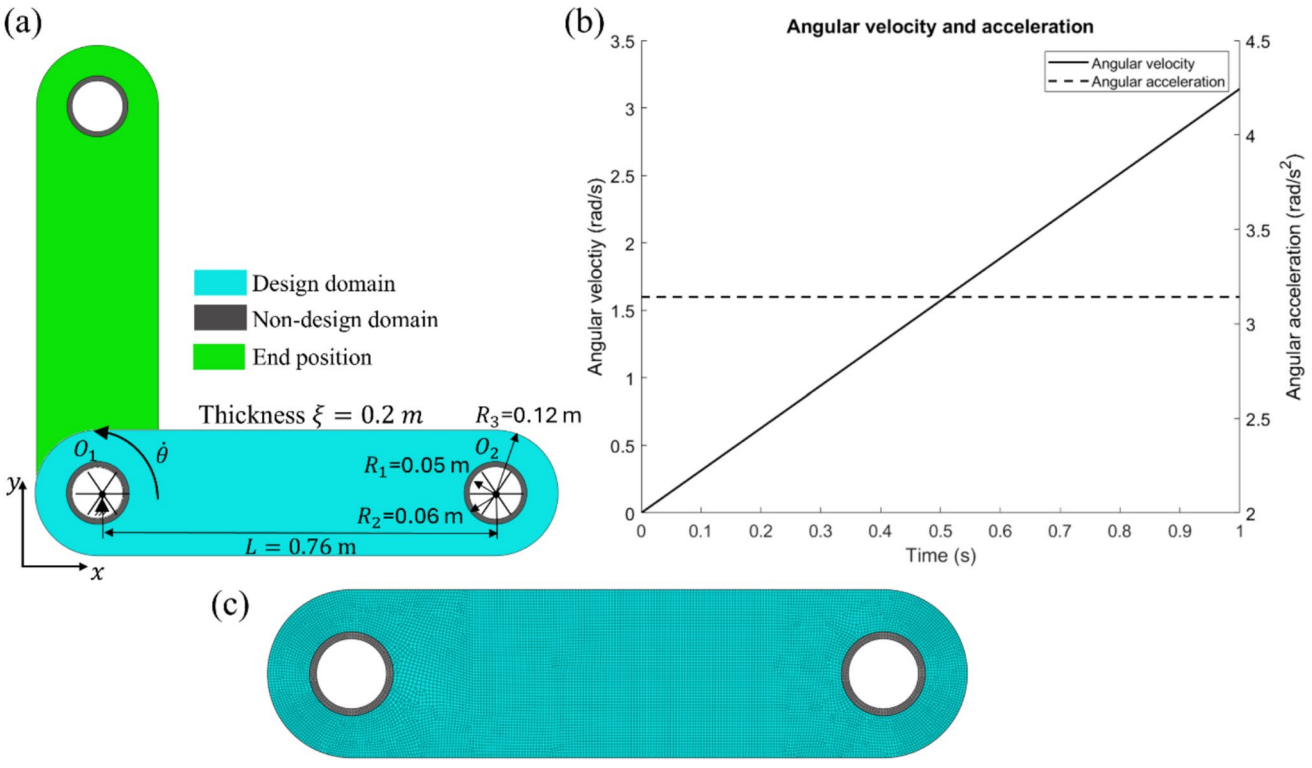

Fig. 3 The example of the robotic arm with one link. **a** Design domain and boundary conditions. **b** The angular velocity and acceleration of the robotic arm. **c** Finite element meshes of the robotic arm

volume fraction during the optimization process. Apparently, the normalized objective function gradually increases from 1 due to the decrease in volume during the optimization process and finally converges to 1.5 after 49 iterations. Figure 4 also shows the topological change of the robotic arm during the optimization process, where the final topology at iteration 49 is similar to the result generated from the cantilever beam (Huang and Xie 2010a, b).

Second, topology optimization is conducted for the robotic arm subject to four time points without design-dependent load, where these four time points correspond to the robotic configurations at $0°, 30°, 60°,$ and $90°$ with respect to the x axis, as depicted in Fig. 3. Figure 5 presents the iteration history of the objective function, volume fraction, and topological evolution during the optimization process. It can be seen that the optimization converges after 44 iterations.

Third, to further investigate the effects of various configurations, topology optimization is performed for the robotic arm by increasing the number of time points from four to ten, but still without design-dependent load. The ten time points are chosen to represent the robotic configurations from $0°$ to $90°$ with the interval of $10°$ with respect to the x axis. Figure 6 shows the iteration history of the objective function and volume fraction during the optimization process.

Fourth, we adopted the proposed design-dependent topology optimization strategy accounting for the robotic arm with time-varying load. Here, the design-dependent load includes two types of loads: self-weight load, which is independent of the robotic motion, and inertial load, which depends on the robotic angular velocity and angular acceleration. To investigate the effects of these two types of loads, we first consider only the self-weight load in the optimization problem. Figure 7 shows the iteration history of the normalized objective function and volume fraction during the optimization process. It can be seen that the objective function first decreases from 1 to 0.92 during the first five iterations due to the self-weight load, then gradually increases from iteration 5 to iteration 30 before satisfying the convergence criteria at iteration 42. Figure 7 also illustrates the topological change of the robotic arm during the optimization process.

Lastly, we perform the proposed design-dependent topology optimization considering time-varying loads with ten time points, where both the self-weight load and the inertial load caused by rotational motion are taken into account. Figure 8 illustrates the iteration history of the objective function and volume fraction. Interestingly, the objective function shows a similar trend to that in Fig. 7 due to the decrease in design-dependent load associated with the deleted elements at the beginning.

Figure 9 exhibits the final topologies obtained from the five different cases for a clearer comparison. Compared to the topology considering only one time point (Fig. 9a), the generated topology of four time points (Fig. 9b) is quite different due to the effect of the load conditions at the four time points considered. Interestingly, the final topology of the

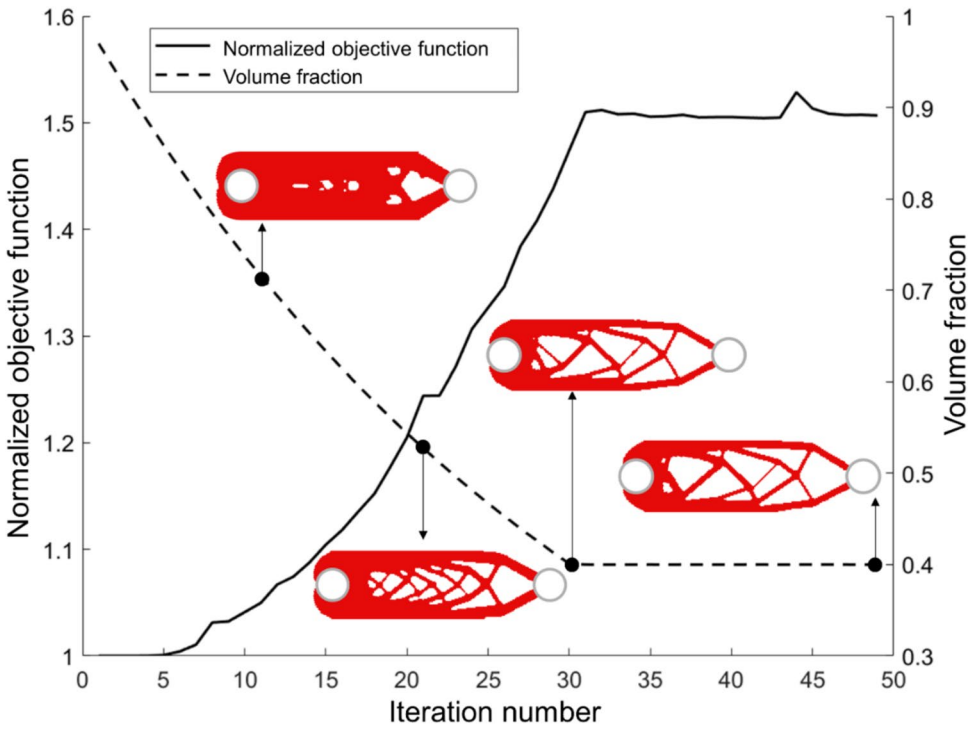

Fig. 4 Iteration history of the conventional topology optimization considering one time point without design-dependent load

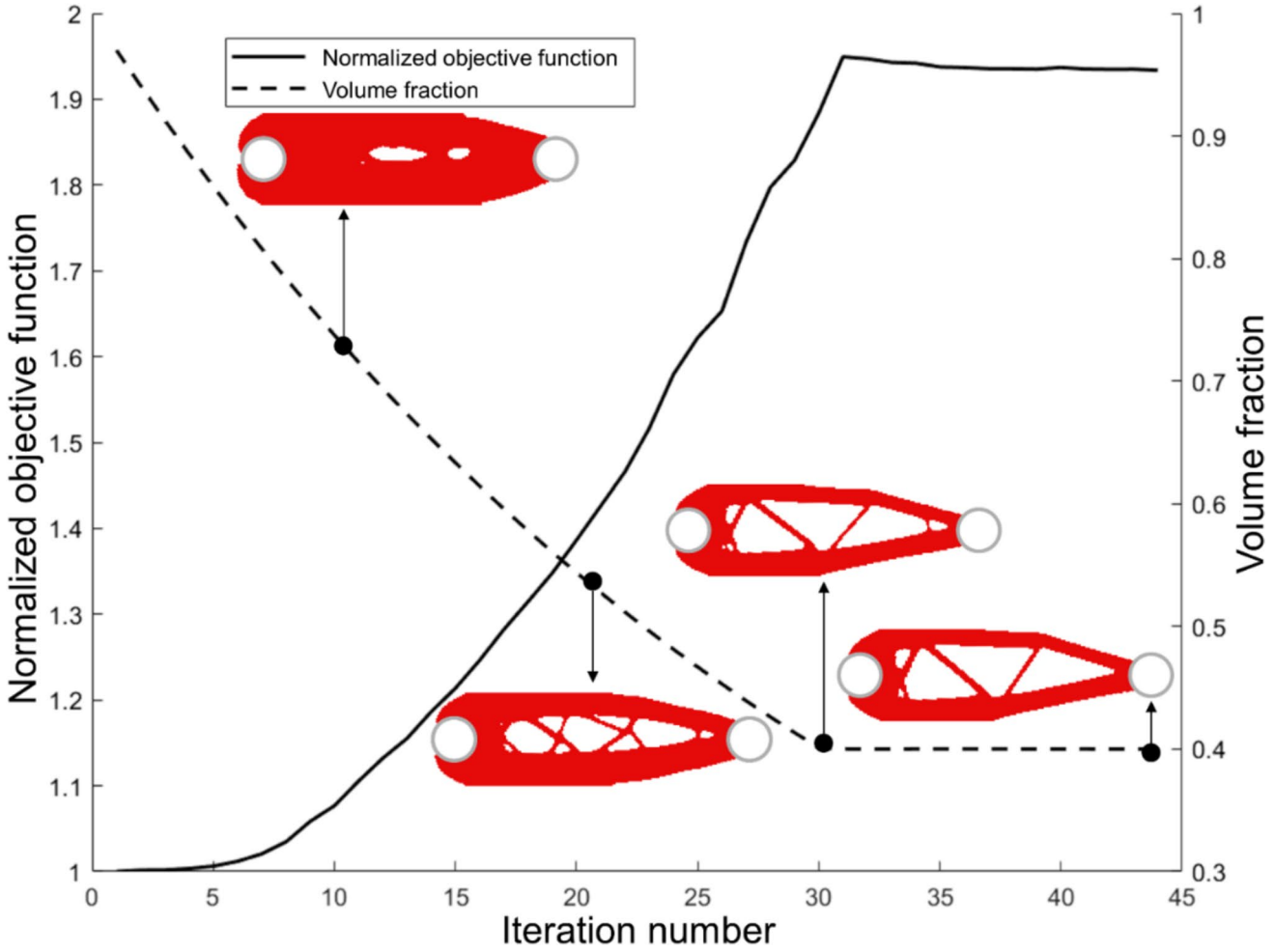

Fig. 5 Iteration history of conventional topology optimization considering four time points without design-dependent load

robotic arm for ten time points (Fig. 9c) is similar to that presented in Fig. 9b, where small differences can be observed in the internal thin struts in Fig. 9c. Although the time points can be further increased to 20, 30, or even more to obtain a more accurate result, the computational cost would be significantly higher for the iterative topology optimization process, which needs further investigation on how to balance them in the future. It is noted that the time integration can be approximated using other numerical approaches, such as the trapezoidal rule to obtain more accurate results.

The final topology of ten time points with self-weight load (Fig. 9d) is significantly different from that in Fig. 9c, demonstrating the necessity of considering design-dependent loads for the design of robotic arms. The final topology of ten time points with both self-weight and inertial load (Fig. 9e) is markedly different from that in Fig. 9d, where more materials tend to be distributed around the rotational center O_1, and the overall structure presents a pattern to overcome the inertial load generated from the rotational movement. The topological design result in Fig. 9e exhibits that inertial load could play a critical role in the design of robotic arms if high-speed rotational movement and large acceleration are involved.

In addition, the compliances of the final optimized topologies were evaluated at ten time points under self-weight and inertial loads to demonstrate the effectiveness of the proposed design. It can be seen that Fig. 9e achieves the minimum compliance with 1.4246 J, 26% lower than that of Fig. 9a, highlighting the necessity of the proposed approach for optimizing high-speed robotic arms.

3.2 Example 2: serial robot with two links

Example 2 employs a serial robot with two links, as illustrated in Fig. 10a, where O_1, O_2, O_3 and O_4 denote the centers of the revolving joints associated with Link 1 and Link 2, respectively. Link 1 and Link 2 use the same initial geometry as illustrated in Fig. 10a. The diameter of

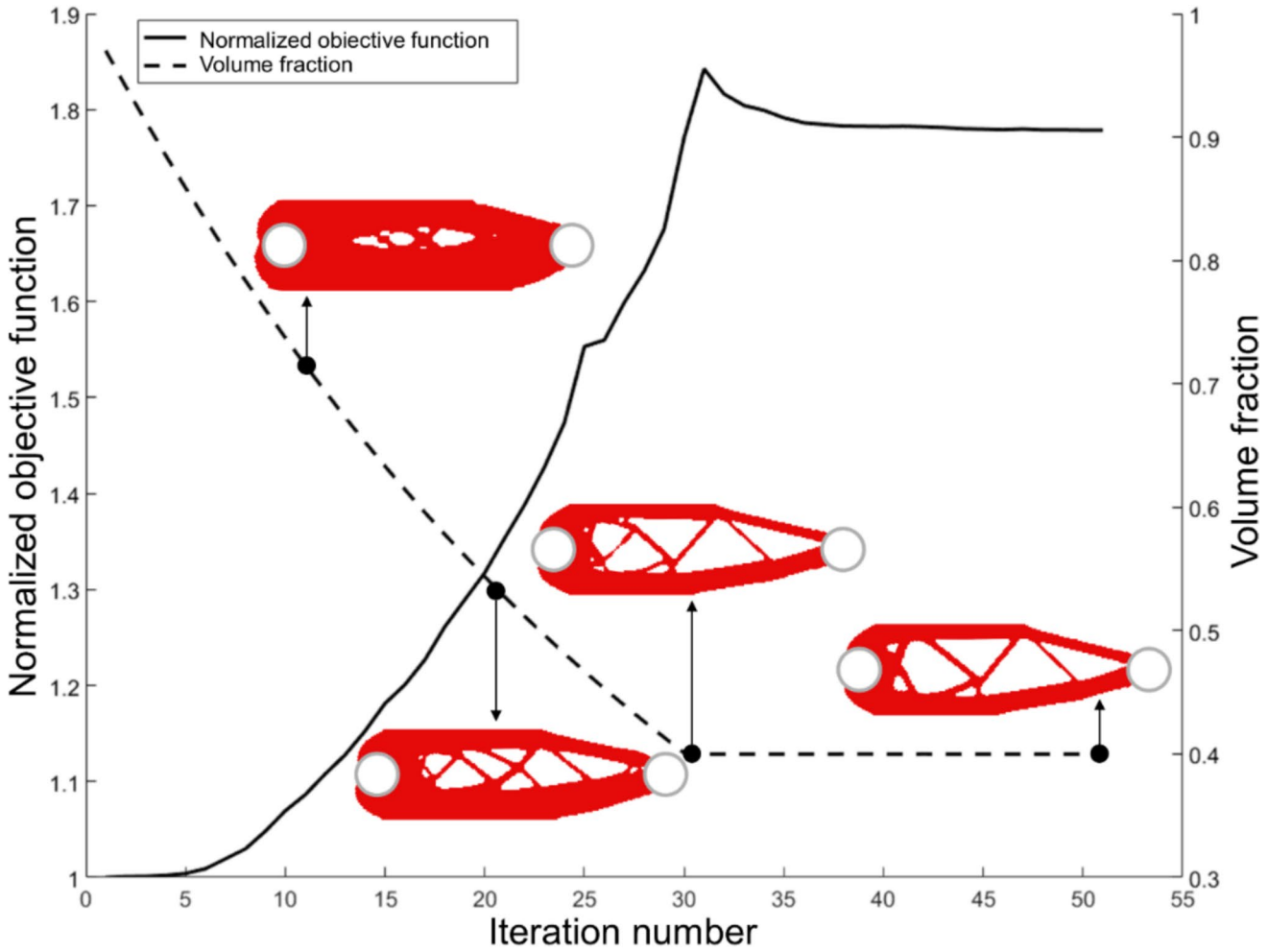

Fig. 6 Iteration history of conventional topology optimization considering ten time points without design-dependent load

the revolving joints ϕ_i is set to 0.1 m, and the distance between the centers of the joints is assumed to be 0.76 m. The radius of the end circular link r equals to 0.12 m, with the thickness $\xi_1 = 0.08$ m for the main structure region and $\xi_2 = 0.04$ m for the joint areas. Similar to Example 1, we separate the robotic links into design domains (colored in bright cyan) and non-design domains (colored in dark gray), as shown in Fig. 10a. The inner and outer diameters of the non-design domain are set to be $\phi_i = 0.1$ m and $\phi_o = 0.12$ m, respectively. Link 1 is meshed with 8-node hexahedral element elements of average size 0.007 m, with a total of 49,949 elements in the design domain and 1,720 elements in the non-design domain. Link 2 has an identical mesh to Link 1. The inner surfaces of those joints are assumed to be fully connected at their centers O_1, O_2, O_3 and O_4.

The serial robot is assumed to have two motor-actuated joints: one rotating about the joint axis at point O_1 with an angular velocity $\dot{\theta}_1 = 90°/\text{s}$, and the other rotating about the joint axis passing through O_2 and O_3 with an angular velocity $\dot{\theta}_2 = 90°/\text{s}$, over a period of 1 s. The motion of the serial robot is provided in Supplementary Movie 2. The end position of the serial robot is illustrated in Fig. 10a as well. The x and y DOFs of center O_1 are constrained, while two concentrated mass points equal to 10 kg each are attached to centers O_2 and O_3 to, respectively, simulate the mass of motor-actuators. A mass point equal to 100 kg is also attached to center O_4 to model the object carried by the robot.

For the topology optimization, the filter radius $\eta_{\min}$ is 0.021 m, equal to three times of the mesh size. The desired total volumes $V^{(1)*}$ and $V^{(2)*}$ of Link 1 and Link 2 are both set to be 30% of the total volume of Link 1 and Link 2 based on full design domain, respectively.

To investigate the effects of different motion case scenarios, two significantly different manners of angular velocity and acceleration, as illustrated in Fig. 10b, are employed. The first motion case scenario involves a linear angular velocity and a constant angular acceleration

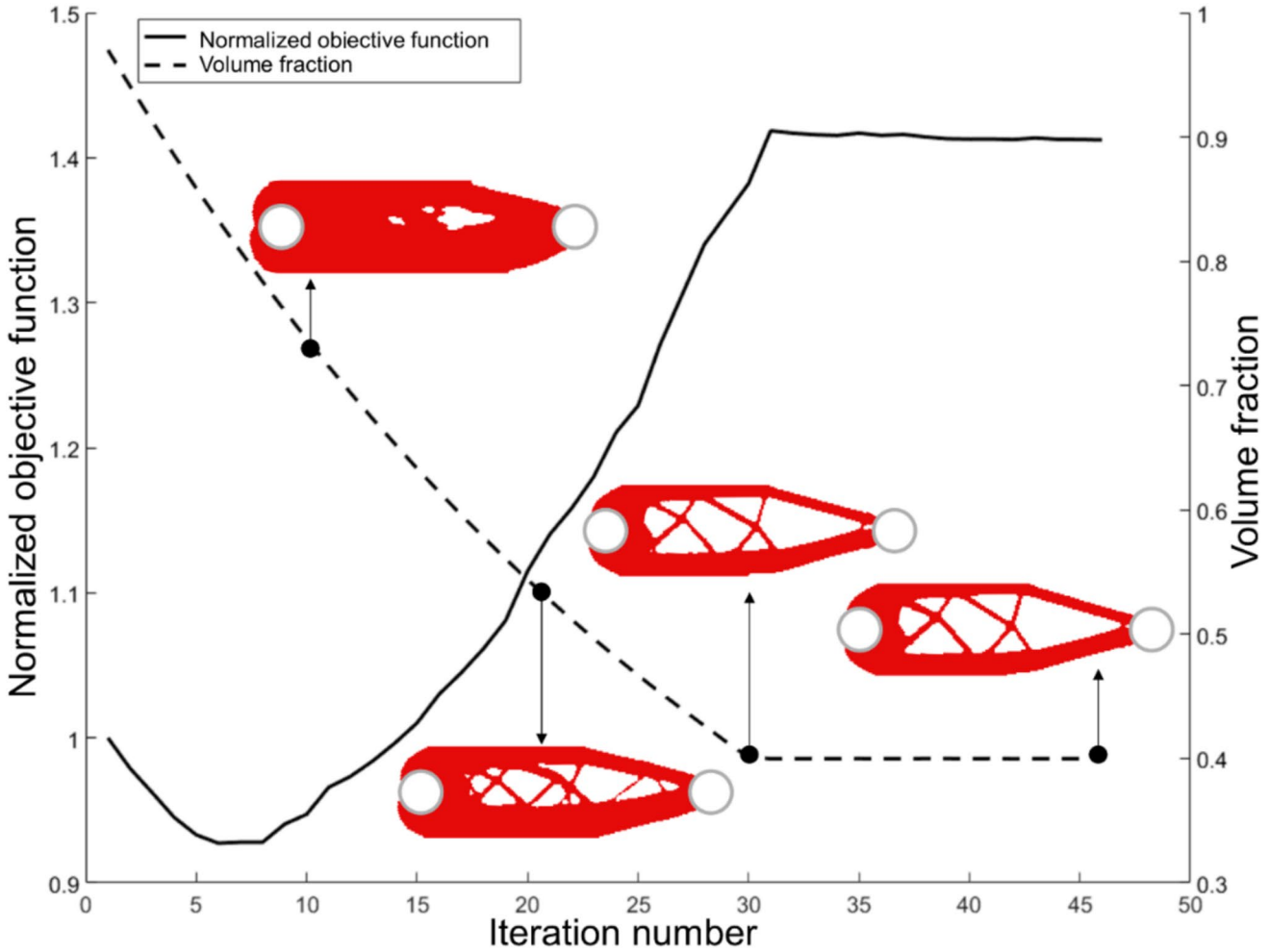

Fig. 7 Iteration history of topology optimization considering ten time points with self-weight load only

similar to those considered in Example 1. In this case, $\ddot{\theta}_1$ and $\ddot{\theta}_2$ represent the angular acceleration of Link 1 with respect to its actuator joint axis and the angular acceleration of Link 2 with respect to its actuator joint axis, respectively, which are assumed to be the same. Thus, we have their linear angular velocity $\dot{\theta}_1 = \dot{\theta}_2 = \pi t$ with respect to the traveling time t. It can be seen that the serial robot would achieve $\theta_1 = \frac{\pi}{2}$ and $\theta_2 = \pi$ when $t = 1$ s, the time elapse from the initial position to the end position, as illustrated in Fig. 10b.

The second motion case scenario involves a nonlinear angular velocity and a nonlinear angular acceleration, where we chose $\ddot{\theta}_1 = \ddot{\theta}_2 = \frac{\pi}{2}\sin(2\pi t)$ rad/s^2 and $\dot{\theta}_1 = \dot{\theta}_2 = -\frac{\pi}{2}\cos(2\pi t) + \frac{\pi}{2}$ rad/s. It can be seen that Link 1 and Link 2 have zero angular acceleration $\ddot{\theta}_1 = \ddot{\theta}_2 = 0$ rad/s^2 when $\tau = 0$ or 1 s, enabling more realistic kinematic behavior at the beginning and end time points.

Similar to Example 1, we take 10 time points to represent different robotic configurations. The 10 time points are chosen to vary θ_1 and θ_2 from 0° to 90° at a uniform interval of 10°.

Figure 11 illustrates the iteration history of the objective function and volume fraction for the optimization problem with the first motion case scenario. It can be seen that the normalized objective function gradually rises from 1 to 1.71 from iteration 1 to iteration 60, converging to around 1.71 at iteration 73. The topological evolution of the two robotic links is also provided in Fig. 11 for a better overview of the optimization process. It can be observed that more materials are present in Link 1 than in Link 2, as Link 1 plays a more crucial role in supporting the whole robotic arms when rotating at a high speed.

For comparison, Fig. 12 illustrates the iteration history of the objective function and volume fraction for the optimization problem with the second motion case scenario. It can be seen that the objective function shows a similar trend to that in Fig. 11, increasing from iteration 1 to iteration

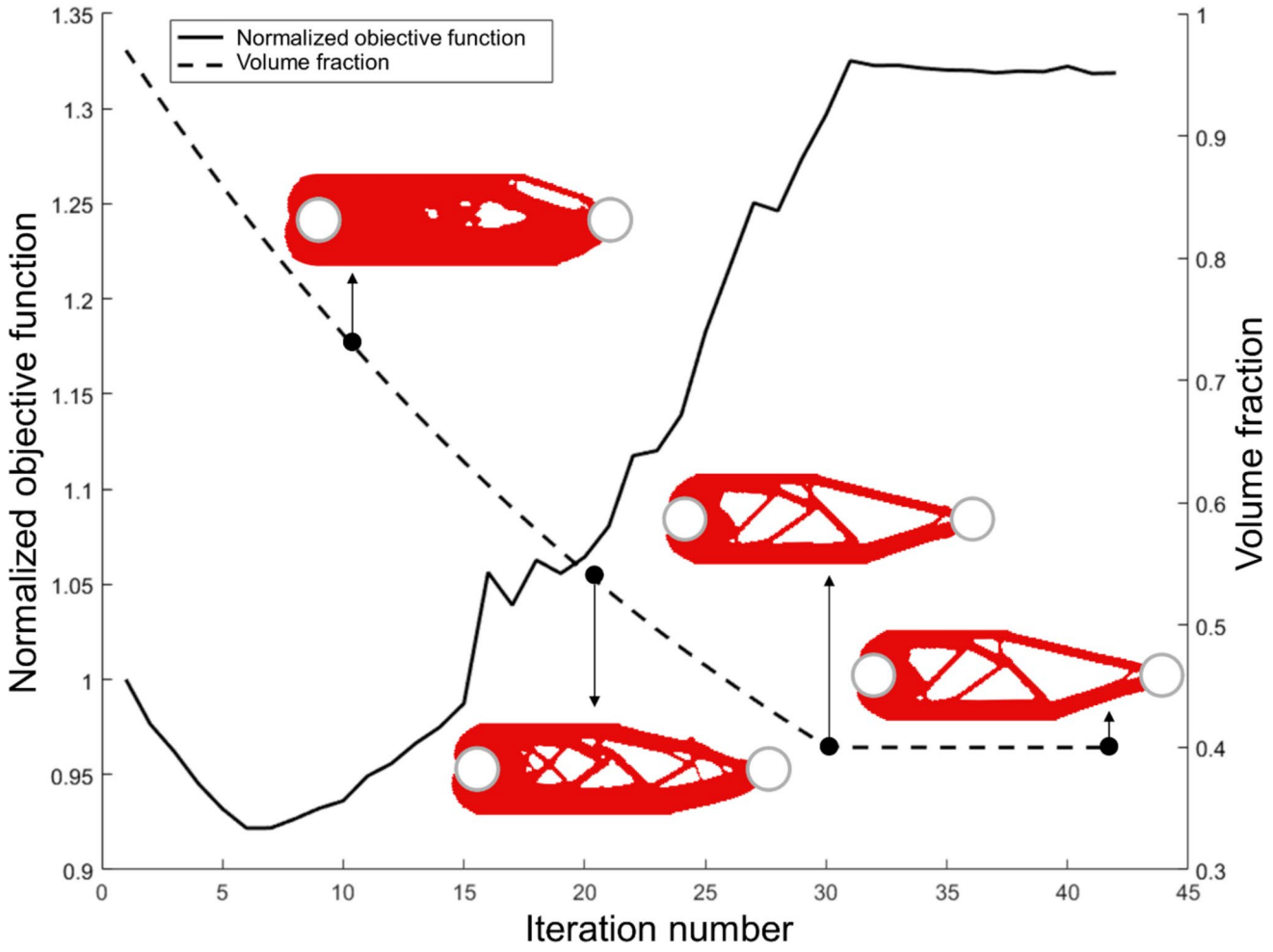

Fig. 8 Iteration history of topology optimization considering ten time points with both self-weight and inertial load

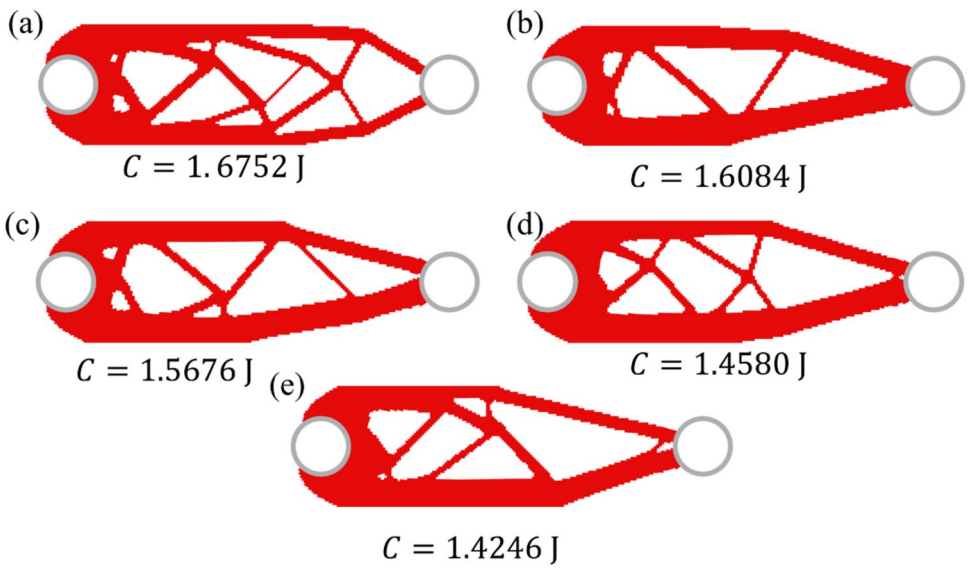

Fig. 9 Final optimized topologies and their compliances subject to ten time points with both self-weight and inertial load. **a** One time point; **b** four time points; **c** ten time points; **d** ten time points with self-weight load; **e** ten time points with self-weight and inertial load

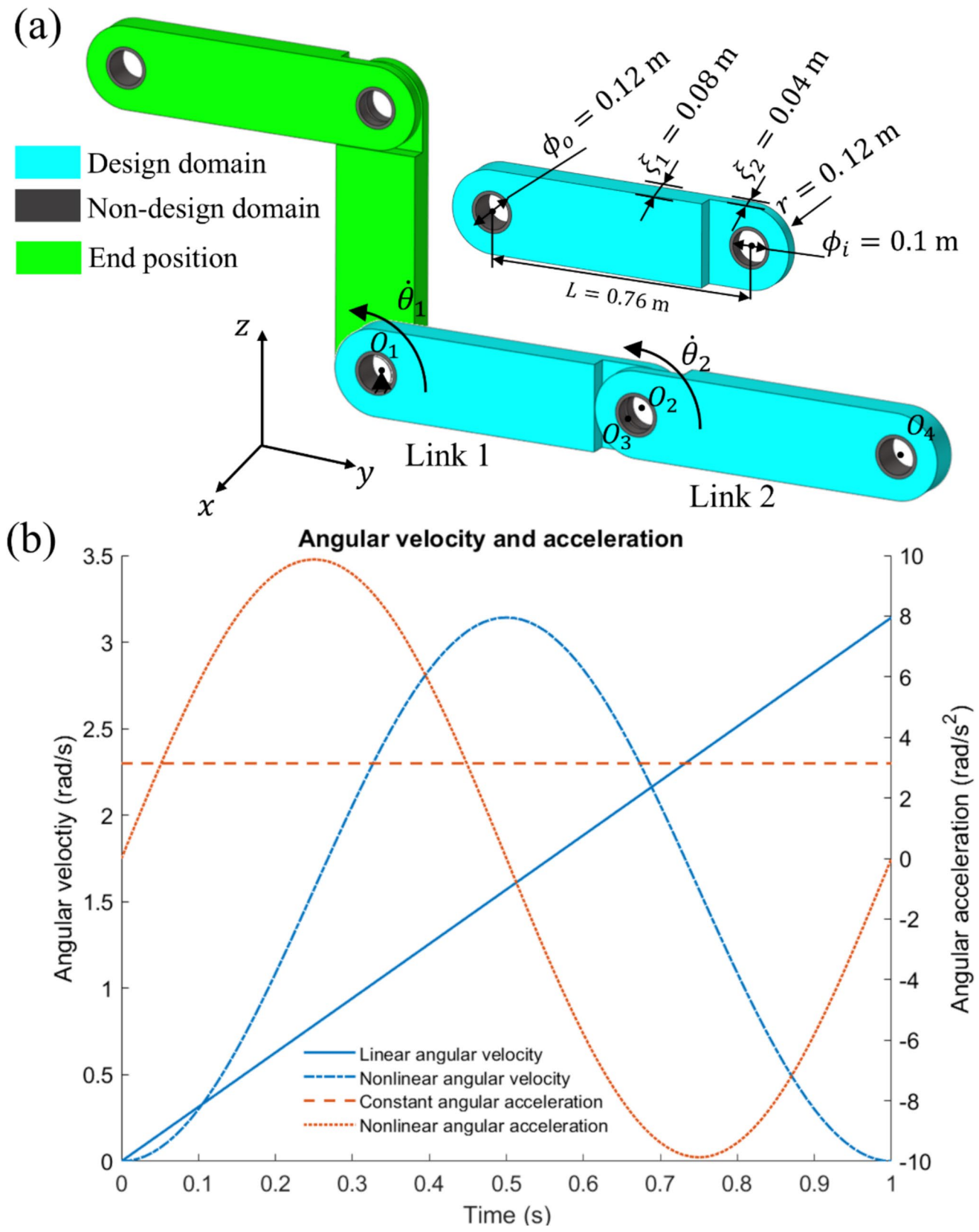

Fig. 10 Example 2—a serial robot with two arms. **a** Design domain and boundary conditions. **b** The linear and nonlinear angular velocity and acceleration of the robotic arms

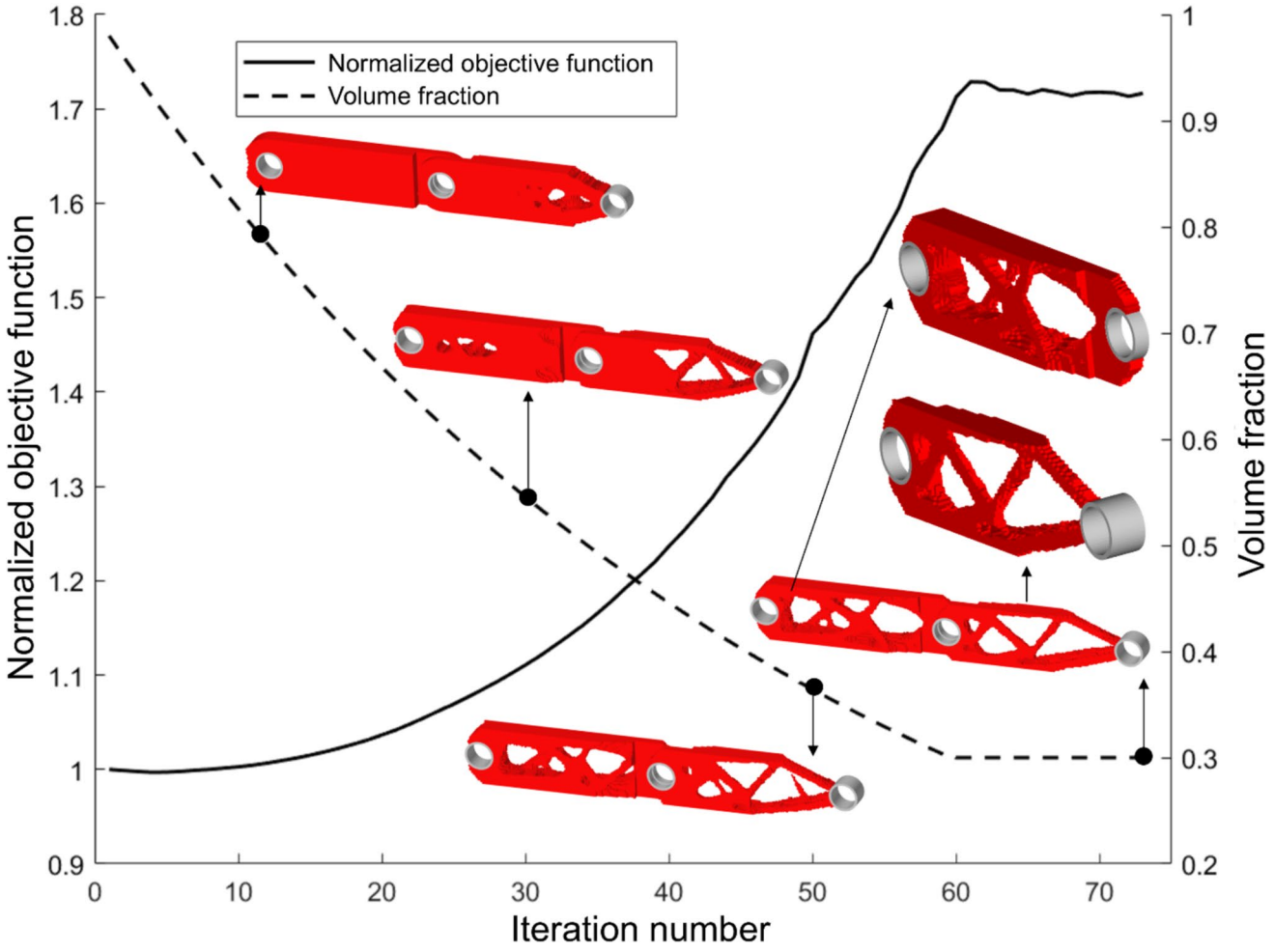

Fig. 11 Iteration history of topology optimization considering ten time points with design-dependent load subject to linear angular velocity and acceleration (Case 1)

60 and converging to 1.71 at iteration 74. In addition, the robotic topologies are presented in Fig. 12 for a better comparison between the two different motion cases. Apparently, the topology of Link 1 in case 2 exhibits a considerable difference from that in case 1 (Fig. 11), while the topology of Link 2 shows a marginal difference from that in Fig. 11. This is because the nonlinear angular acceleration and velocity dramatically change the inertial load from $t = 0$ s to $t = 1$ s. Furthermore, the design-dependent inertial load could accumulate its effect from the far end to the rotational center O_1, thus leading to a notable impact on the topology of Link 1. Through the comparative studies, it can be noted that the motion strategy of the serial robot could play a critical role in the resultant topologies, demonstrating the importance of coupling the design-dependent load with a specific motion strategy adopted by the robot.

3.3 Example 3: 3D spatial three-link robot

Example 3 focuses on a serial robot equipped with three links rotating in 3D space, as shown in Fig. 13a. The three robotic links are assumed to have identical initial geometries, with the dimensions illustrated in Fig. 13a. The diameter of the revolving joints ϕ_i is set to 0.05 m. The radius r of Link 1 equals 0.05 m. The lengths of L_1 and L_2 are set to 0.102 m and 0.246 m. The total size along the global x - axis direction L_4 is 0.1 m and L_3 is 0.055 m. O_1, O_2, O_3, O_4, O_5 and O_6 represent the centers of the joints of Link 1, Link 2, and Link 3, respectively, as shown in Fig. 13a. Each link is divided into a design domain (colored in bright cyan) and a non-design domain (colored in dark gray). The inner and outer diameters of the non-design domain are equal to $\phi_i = 0.05$ m and $\phi_o = 0.06$ m, respectively. Link 1 is meshed with 8-node hexahedral elements of average size 0.004 m. The design domain and non-design domain have 49,167 and

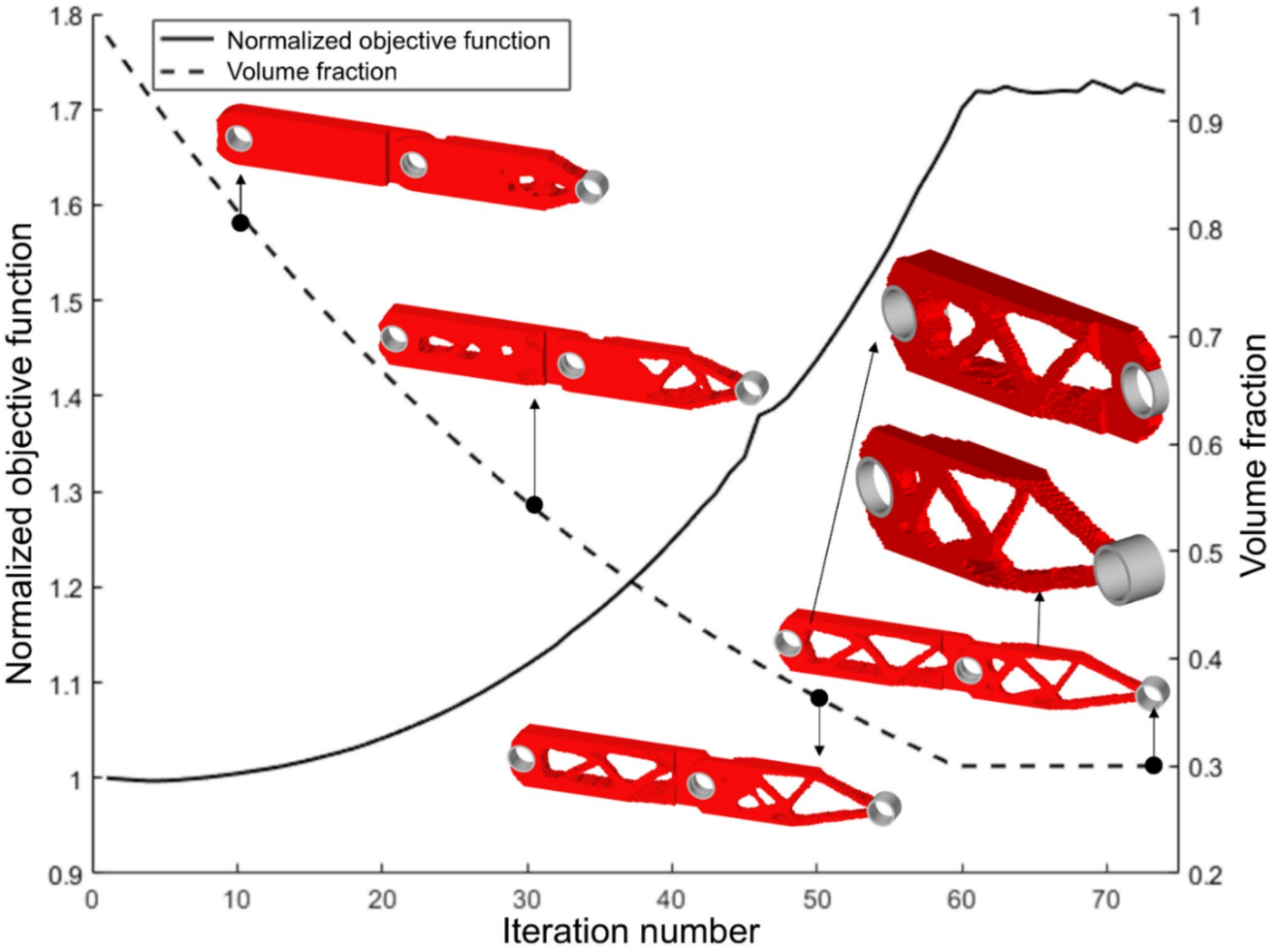

Fig. 12 Iteration history of topology optimization considering 10 time points with design-dependent loads subject to nonlinear angular velocity and acceleration (Case 2)

1,958 elements, respectively. Links 2 and 3 share the same mesh as Link 1.

The serial robot is assumed to have three motor-actuators that rotate around the joint axis via center O_1, the joint axis through O_2 and O_3, and the joint axis through O_4 and O_5 for 90° as illustrated in Fig. 13a. The 3D motion of the robot is illustrated in Supplementary Movie 3. Concentrated mass points of 10 kg are attached to O_2, O_3, O_4 , and O_5 to simulate the mass of motor-actuators. Additionally, a concentrated mass point of 100 kg is associated with center O_6 to model the object carried by the robot. The inner surfaces of the joints are assumed to be fully connected to their centers in FE analyses. The filter radius $\eta_{min} = 0.012$ m is set to be three times the mesh size. The desired volume V^* is set to be 20% of the full design domain of all three arms.

In this example, we employ the nonlinear angular velocity $\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3$ and nonlinear angular acceleration for $\ddot{\theta}_1, \ddot{\theta}_2, \ddot{\theta}_3$ for the joints at center O_1, center O_2, O_3, and center O_4, O_5. Two motion case scenarios are considered with the same nonlinear pattern but different traveling time here. The first motion case is the same as that defined in Example 2, with $\dot{\theta}_1 = \dot{\theta}_2 = \dot{\theta}_3 = -\frac{\pi}{2}\cos(2\pi t) + \frac{\pi}{2}$ rad/s , and $\ddot{\theta}_1 = \ddot{\theta}_2 = \ddot{\theta}_3 = \frac{\pi}{2}\sin(2\pi t)$ rad/s^2. In this case, the robot takes $T = 1$ s to travel from its initial position to the end position. For the second motion case scenario, the same nonlinear pattern remains but the traveling time increases from $T = 1$ s to $T = 5$ s. The deduced angular acceleration is then $\ddot{\theta}_1 = \ddot{\theta}_2 = \ddot{\theta}_3 = \frac{\pi^2}{25}\sin\left(\frac{2\pi}{5}t\right)$ rad/s^2, and the angular velocity $\dot{\theta}_1 = \dot{\theta}_2 = \dot{\theta}_3 = -\frac{\pi}{10}\cos(2\pi t) + \frac{\pi}{10}$ rad/s case 2 here. It can be noted that $\ddot{\theta}_1 = \ddot{\theta}_2 = \ddot{\theta}_3 = 0$ rad/s^2 and $\dot{\theta}_1 = \dot{\theta}_2 = \dot{\theta}_3 = 0$rad/s when $t = 0$or5s as plotted in Fig. 13b. Moreover, $\theta_1 = \theta_2 = \theta_3 = \frac{\pi}{2}$ rad when $t = 5$ s,

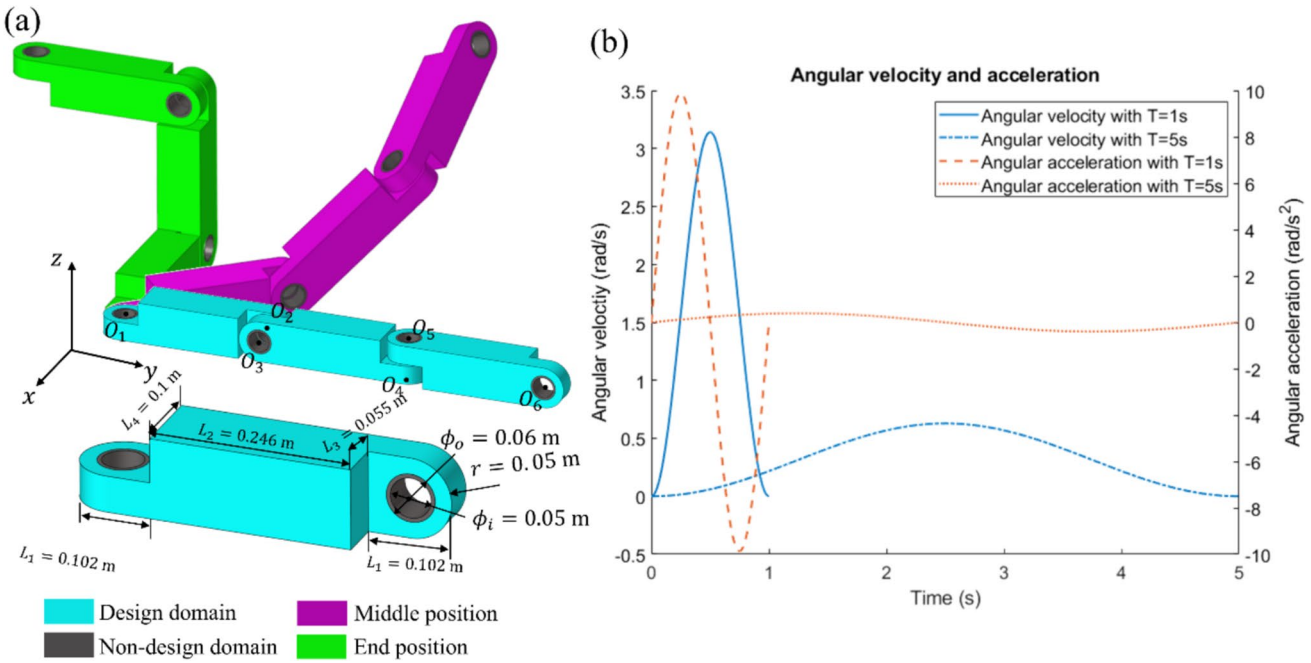

Fig. 13 The 3D spatial serial robot with three links. **a** Design domain and boundary conditions. **b** The angular velocity and acceleration with time periods of 1 s and 5 s (Example 3)

denoting that the robot takes 5 s to travel from its start position to the end position.

Figure 14 plots the history of the topology optimization for a traveling time $T = 1$ s. Clearly, the normalized objective function increases from iteration 1 to iteration 80 when the desired volume fraction of 20% is achieved. After that, the objective function converges to 2.80 times the initial mean compliance after 111 iterations. The topological evolution of the three robotic arms is provided in Fig. 14. Similar to the topology obtained in Example 2, the final topology in Fig. 14 exhibits that more materials are allocated to arm 1 and arm 2, while less material is required for arm 3 due to the rotational movement of the serial robot. Moreover, all the arms present internal hollow structures, which are reasonable for improving the efficiency of material usage from an empirical point of view.

Figure 15 shows the iteration history of the topology optimization for a traveling time $T = 5$ s. The normalized objective function and volume fraction exhibit the same trend as those in Fig. 14. Nevertheless, significant differences in the topology evolution can be observed during the optimization process. In the final topology, arm 1 presents a significantly different layout from that in Fig. 14. In addition, the structure of Link 2 shows a marginal difference from that in Fig. 14, while Link 3 remains almost the same subject to different traveling time. It can be observed that the various inertial loads caused by the change in traveling time have a notable impact on the structural layout of the serial robot but are marginally weaker from the base to the far end of the robotic arms. It is also noted that the compliance of the design with the traveling time $T = 1$ s simulated over 1 s is higher than that with $T = 5$ s simulated over 5 s (15.24 J vs 14.98 J). Although both cases use the same number of time points (10) and the same robotic configuration at each time point, their inertial loads differ significantly due to variations in angular velocity and acceleration. Specifically, the longer traveling time ($T = 5$ s) results in lower operational speed, which leads to reduced inertial loads, and consequently a lower compliance value. The comparative studies between different traveling times demonstrate that the operational time of robots is also an important factor that should be considered in the topological design.

4 Conclusions

In this study, we developed a design-dependent topology optimization framework for robotic arms considering time-varying configurations. The BESO method was employed to account for design-dependent loads, including self-weight and inertial loads that a robot experiences during high-speed rotational motion. We outlined the transient BESO framework and the finite element (FE) analysis involving

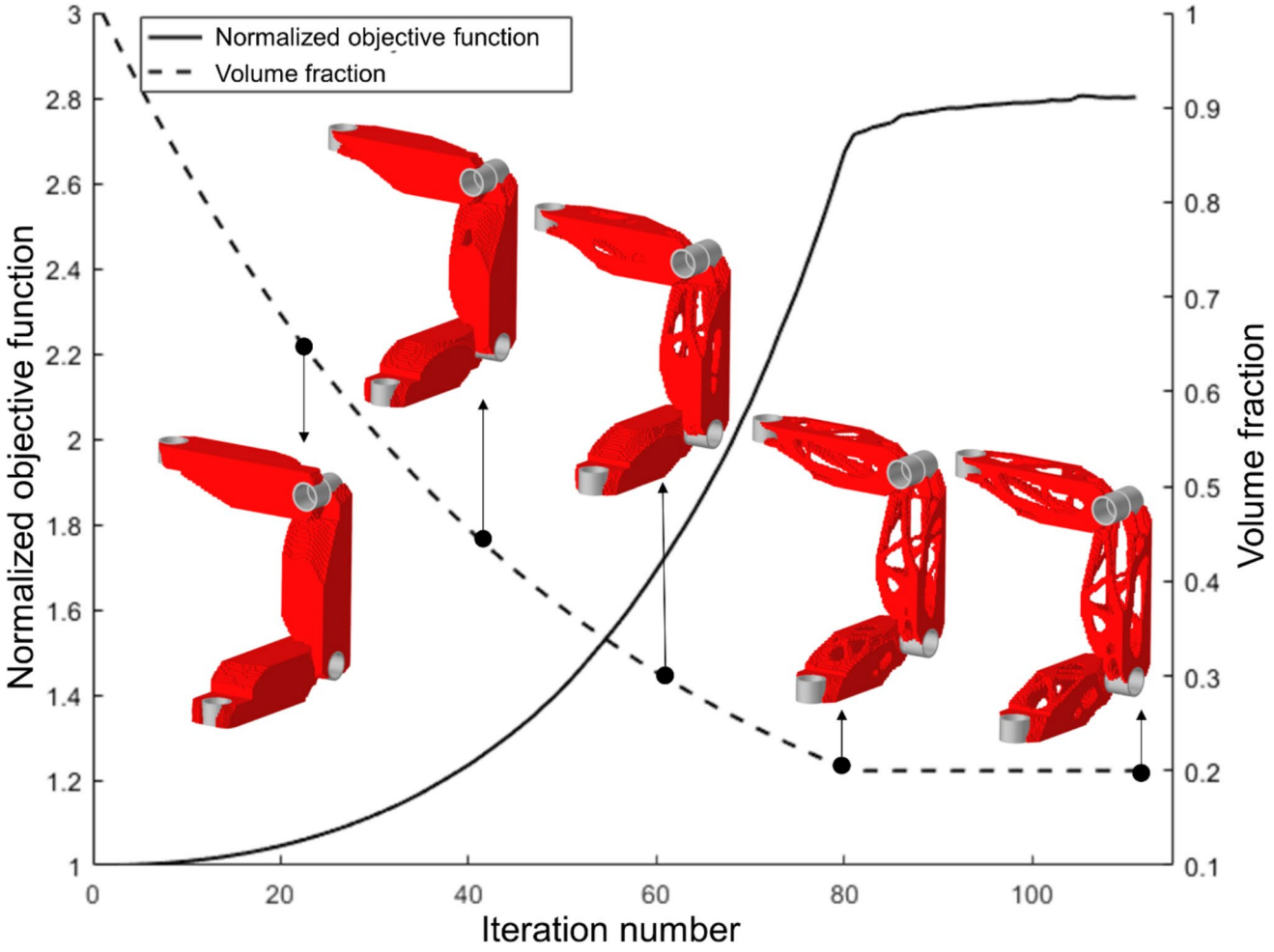

Fig. 14 Iteration history of topology optimization considering ten time points with design-dependent load subject to a time period of $T = 1$ s (Example 3—Case 1)

design-dependent loads. To calculate the inertial load, we computed the acceleration of serial robots with multiple arms in a forward manner. Sensitivity analysis was performed to provide ranking information for the BESO method to efficiently add and/or remove elements in an evolutionary way.

Three numerical examples were investigated, including a 2D single robotic arm, a 3D model with two serial arms rotating in a plane, and a 3D model with three serial arms exhibiting spatial movement. Comparative studies in each example showcased the importance of design-dependent loads subject to various time points on the generated topologies. Additionally, typical motion strategies and different traveling times were investigated to illustrate their influence on the resultant topological design.

The proposed design-dependent transient approach sheds new light on the applications of topology optimization for robots, comprehensively considering various key factors from their movements. Moreover, future work is anticipated to integrate the proposed approach with control strategies and time-dependent dynamics, forming a more practical tool for the coupled optimization of structural performance and movement control.

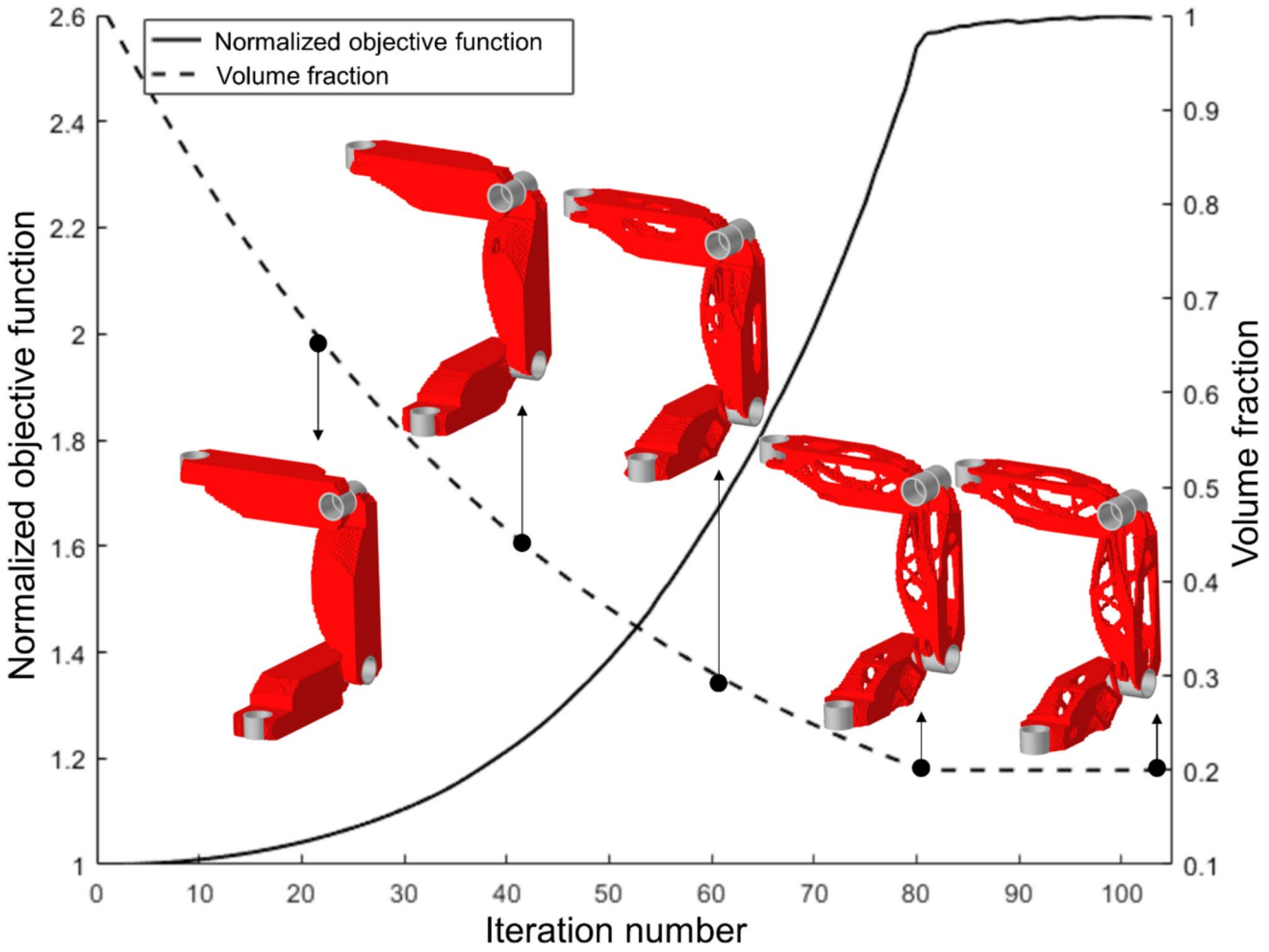

Fig. 15 Iteration history of topology optimization considering ten time points with design-dependent load subject to a time period of $t = 5$ s (Example 3—Case 2)

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00158-025-04129-1.

Author contributions Chi Wu contributed toward conceptualization, methods, data collection, writing-original draft, and review & editing. Yanan Xu contributed toward methods and analysis. Jianguang Fang contributed toward conceptualization and writing-review & editing. Guangyong Sun contributed toward writing-review & editing. Grant P Steven contributed toward conceptualization, writing-review & editing, supervision, and funding acquisition. Qing Li contributed toward conceptualization, writing-review & editing, supervision, and funding acquisition.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions. This work is supported by the Australian Research Council (ARC) through the Discovery project (DP230103180).

Data availability All the results contained within this manuscript are available from the corresponding author upon request.

Declarations

Conflict of interest The authors declare no conflict of interests.

Replication of results All numerical results can be reproduced with the presented methods in Sects. 2 and 3. Data and materials are available upon request to the corresponding author.

References

Abaqus G (2016) Abaqus 6.16, vol 3. Dassault Systemes Simulia Corporation, Providence, p 73

Alkalla MG, Fanni MA (2021) Integrated structure/control design of high-speed flexible robot arms using topology optimization. Mech Based des Struct Mach 49(3):381–402
Alkhafaji FSM (2021) Modeling and control high speed robotic arm for industrial applications. In: Modeling and control high speed robotic arm for industrial applications. IEEE, pp 121–130
Allaire G, Jouve F, Toader A-M (2004) Structural optimization using sensitivity analysis and a level-set method. J Comput Phys 194(1):363–393
Bendsøe MP, Sigmund O (1999) Material interpolation schemes in topology optimization. Arch Appl Mech 69:635–654
Bowling AP, Kim C (2006) Velocity effects on robotic manipulator dynamic performance. J Mech des. https://doi.org/10.1115/1.2336255
Bruyneel M, Duysinx P (2005) Note on topology optimization of continuum structures including self-weight. Struct Multidisc Optim 29:245–256
Canh TN, Duc ST, The HN, Dao TH, HoangVan X (2024) Optimal design and fabrication of frame structure for dual-arm service robots: an effective approach for human–robot interaction. Eng Sci Technol Int J 56:101763
Chong Z, Xie F, Liu X-J, Wang J, Li P (2020) Worst case identification based topology optimization of a 2-DoF hybrid robotic arm. Int J Intell Robot Appl 4(2):136–148
Dagnino G, Kundrat D (2024) Robot-assistive minimally invasive surgery: trends and future directions. Int J Intell Robot Appl. https://doi.org/10.1007/s41315-024-00341-2
Deng S, Wang Y, Cai H, Li K, Yang F, Wang Y (2018) Structural topology optimization research for a six-DOF space robotic manipulator. In: Structural topology optimization research for a six-DOF space robotic manipulator. Atlantis Press, pp 543–548
Gao T, Zhang W (2010) Topology optimization involving thermo-elastic stress loads. Struct Multidisc Optim 42:725–738
Han Y, Xu B, Duan Z, Huang X (2022) Stress-based bi-directional evolutionary structural topology optimization considering nonlinear continuum damage. Comput Methods Appl Mech Eng 396:115086
Huang X, Xie Y (2008a) A new look at ESO and BESO optimization methods. Struct Multidisc Optim 35:89–92
Huang X, Xie Y (2008b) Topology optimization of nonlinear structures under displacement loading. Eng Struct 30(7):2057–2068
Huang X, Xie M (2010a) Evolutionary topology optimization of continuum structures: methods and applications. Wiley
Huang X, Xie Y (2010b) Evolutionary topology optimization of continuum structures with an additional displacement constraint. Struct Multidisc Optim 40:409–416
Huang X, Xie Y (2011) Evolutionary topology optimization of continuum structures including design-dependent self-weight loads. Finite Elem Anal des 47(8):942–948
Huang X, Zuo Z, Xie Y (2010) Evolutionary topological optimization of vibrating continuum structures for natural frequencies. Comput Struct 88(5–6):357–364
Inc., T.M. (2022) MATLAB version: 9.13.0 (R2022b). The MathWorks Inc.
Kishida M, Kurahashi T (2023) Development of a novel non-positive definite correspondence modified optimality criteria method for multi-objective density-based topology optimization. Mech Eng J 10(6):23-00353-00323–00353
Křepelka J, Schovánek P, Tuček P, Hrabovský M, Jáně F (2024) Optimization of component assembly in automotive industry. Meas Sci Rev 24(1):36–41
Kuruvilla JK, Seth A, Duttagupta J, Sharma S, Jaiswal A (2024) Structural design and analysis of 6-DOF cylindrical robotic manipulators for automated agriculture. In: From precision agriculture for sustainability. Apple Academic Press, pp 147–168
Lattanzi L, Cristalli C, Massa D, Boria S, Lépine P, Pellicciari M (2020) Geometrical calibration of a 6-axis robotic arm for high accuracy manufacturing task. Int J Adv Manuf Technol 111(7):1813–1829
Lee D, Seo T (2017) Lightweight multi-DOF manipulator with wire-driven gravity compensation mechanism. IEEE ASME Trans Mechatron 22(3):1308–1314
Liu K, Paulino GH (2019) Tensegrity topology optimization by force maximization on arbitrary ground structures. Struct Multidisc Optim 59:2041–2062
Liu B, Sha L, Huang K, Zhang W, Yang H (2022) A topology optimization method for collaborative robot lightweight design based on orthogonal experiment and its applications. Int J Adv Robot Syst. https://doi.org/10.1177/17298814211056143
Lou G, Wang C, Xu Z, Liang J, Zhou Y (2024) Controlling soft robotic arms using hybrid modelling and reinforcement learning. IEEE Robot Autom Lett. https://doi.org/10.1109/LRA.2024.3418312
Lu Z, Chauhan A, Silva F, Lopes LS (2012) A brief survey of commercial robotic arms for research on manipulation. In: A brief survey of commercial robotic arms for research on manipulation. IEEE, pp 986–991
Park G-J (2011) Technical overview of the equivalent static loads method for non-linear static response structural optimization. Struct Multidisc Optim 43(3):319–337
Paska Z, Rojicek J, Ferfecki P, Fusek M, Heczko D, Krys V (2020) Methodology of arm design for mobile robot manipulator using topological optimization. MM Sci J 2:3918–3925
Querin OM, Steven GP, Xie YM (1998) Evolutionary structural optimisation (ESO) using a bidirectional algorithm. Eng Comput 15(8):1031–1048
Reddy JN (1993) An introduction to the finite element method. N Y 27(14):348–349
Shi H, Li R, Bai X, Zhang Y, Min L, Wang D, Lu X, Yan Y, Lei Y (2023) A review for control theory and condition monitoring on construction robots. J Field Robot 40(4):934–954
Srinivas GL, Javed A (2020a) Multi-body dynamic optimization for upper arm of industrial manipulator. In: Multi-body dynamic optimization for upper arm of industrial manipulator
Srinivas GL, Javed A (2020b) Topology optimization of rigid-links for industrial manipulator considering dynamic loading conditions. Mech Mach Theory 153:103979
Srinivas GL, Javed A (2021) Topology optimization of KUKA KR16 industrial robot using equivalent static load method. In: Topology optimization of KUKA KR16 industrial robot using equivalent static load method, pp 1–6
Stolpe M, Svanberg K (2001) An alternative interpolation scheme for minimum compliance topology optimization. Struct Multidisc Optim 22:116–124
Vicentini F (2021) Collaborative robotics: a survey. J Mech des 143(4):040802
Wang MY, Wang X, Guo D (2003) A level set method for structural topology optimization. Comput Methods Appl Mech Eng 192(1–2):227–246
Wu C, Gao Y, Fang J, Lund E, Li Q (2017) Discrete topology optimization of ply orientation for a carbon fiber reinforced plastic (CFRP) laminate vehicle door. Mater des 128:9–19
Wu C, Fang J, Li Q (2019) Multi-material topology optimization for thermal buckling criteria. Comput Methods Appl Mech Eng 346:1136–1155
Wu C, Zheng KK, Fang JG, Steven GP, Li Q (2020) Time-dependent topology optimization of bone plates considering bone remodeling. Comput Methods Appl Mech Eng 359:112702
Xia L, Zhang L, Xia Q, Shi T (2018) Stress-based topology optimization using bi-directional evolutionary structural optimization method. Comput Methods Appl Mech Eng 333:356–370
Xie YM, Steven GP (1993) A simple evolutionary procedure for structural optimization. Comput Struct 49(5):885–896

Xie F, Gao Y, Pan T, Gao D, Wang L, Xu Y, Wu C (2023) Novel lightweight connecting bracket design with multiple performance constraints based on optimization and verification process. Electron Res Arch. https://doi.org/10.3934/era.2023104

Xu H, Guan L, Chen X, Wang L (2013) Guide-weight method for topology optimization of continuum structures including body forces. Finite Elem Anal des 75:38–49

Yin H, Liu J, Yang F (2019) Hybrid structure design of lightweight robotic arms based on carbon fiber reinforced plastic and aluminum alloy. IEEE Access 7:64932–64945

Yoshida T, Onishi Y, Kawahara T, Fukao T (2022) Automated harvesting by a dual-arm fruit harvesting robot. Robomech J 9(1):19

Yue X-H, Yin X, Sun Z-Y, Liu L-Y, Wang Y, Xu G-K, Cao C, Zhang L-Y (2024) Flexible, lightweight, tunable robotic arms enabled by X-tensegrity inspired structures. Compos Struct. https://doi.org/10.1016/j.compstruct.2024.118331

Zhang W, Yuan J, Zhang J, Guo X (2016) A new topology optimization approach based on moving morphable components (MMC) and the ersatz material model. Struct Multidisc Optim 53:1243–1260

Zhong D, Wang K, Zuo Y, Wei M, Xiong J, Wang H, Zhang P, Shang N, Chen Z, Pei P (2023) Metal–air batteries for powering robots. J Mater Chem A 11(46):25115–25135