Journal of Intelligent & Robotic Systems (2024) 110:23
https://doi.org/10.1007/s10846-024-02050-1

REGULAR PAPER

Online Capability Based Task Allocation of Cooperative Manipulators

Keshab Patra[1] · Arpita Sinha[2] · Anirban Guha[1]

Received: 29 May 2023 / Accepted: 2 January 2024 / Published online: 25 January 2024

Abstract
The cooperative manipulator group can accomplish complex and heavy payload tasks of object manipulation and transportation compared to a single manipulator. Effective coordination is crucial for cooperative task accomplishments. Multi-manipulator task distribution is highly complex because of the varying dynamic capabilities of the manipulators. We have introduced a novel fastest technique to quantify the dynamic task capability of the cooperative manipulator by scalar quantity and allocate the task accordingly. The scalar quantity determines the capability of applying an external wrench by end effector (EE) in line with the required wrench at the center of mass of the manipulating object. This quantity helps to diminish tracking errors in object manipulations or transportation and actuator saturation avoidance. The task distribution among the members is in proportion to their computed dynamic capability to ensure equal priority to the individual manipulators. The proposed task distribution formulation ensures the minimum magnitude of wrench interaction at the grasp point and the minimum internal wrench build-up in the object. Several physical simulation results assure trajectory tracking performance with the proposed task capability metric. The same metric aids in identifying the least capable manipulator, rearranging members for better performance, and deciding the required number of manipulators in the manipulator group.

Keywords Cooperative manipulation · Task allocation · Wrench allocation · Dynamic capability

1 Introduction

Robotic systems have become ingrained in manufacturing, remote exploration, and other areas requiring dirty, dangerous, or dull tasks. Sometimes, a single robot can only accomplish the desired task if it becomes prohibitively large and expensive. Examples of such tasks include handling an object that requires too many degrees of freedom, transporting heavy or oversized payload, and multipart assembly without a fixture. Cooperative object manipulation is resorted to when the task is beyond the capability of an individual manipulator. The cooperative manipulators enhance the entire system's coverage, flexibility, and redundancy. This advantage comes with the cost of complexity in robot coordination, communication, and task allocation, especially while manipulating an unknown object in a cluttered environment. Numerous combinations of task allocation to the individual manipulator result in the same wrench applied to the manipulated object. The wrench allocation scheme optimizes the capability utilization of the manipulator, exploiting the inherent redundancy in cooperative manipulations.

Wrench allocation approaches are reported in [1–6] that considers constant collaborative task sharing without concern to the varying task capability of manipulators. Task allocation without capability measures results in non-optimal cooperation, significant tracking error, and internal wrench. The capability measure for cooperative manipulators is also necessary for the optimal utilization of the cooperative group and to know the capacity margin for disturbance and uncertainty mitigation.

In previous work for task capability measures, a vertex-search algorithm to compute the task-space force polytope's vertices has been introduced in [7, 8] using slack variables along with singular value decomposition methods. The

✉ Anirban Guha
anirbanguha1@gmail.com

Keshab Patra
keshabpatra19@gmail.com

Arpita Sinha
arpita.sinha@iitb.ac.in

[1] Department of Mechanical Engineering, Indian Institute of Technology Bombay, Powai, Mumbai 400076, Maharashtra, India

[2] System & Control Engineering, Indian Institute of Technology Bombay, Powai, Mumbai 400076, Maharashtra, India

algorithm has further improved in [9, 10] by reducing the computational complexity using geometric characteristics of the joint torque space. Though the algorithm notably reduces complexity and computation time, it does not allow real-time execution in cooperative manipulations. A detailed literature survey has been represented in Section 2.

We introduce a novel online task capability computation algorithm for manipulators that is computationally faster and simpler than the latest algorithm [9, 10], considering the manipulator arm dynamics and the joint actuator limits.

The main contribution of this work is as follows:

1. *Task capability measures:* We propose a real-time implementable task capability finding algorithm for individual and cooperative manipulator groups considering the manipulator dynamics and the joint actuator limits.
2. *Online task allocation:* Formulated an online task allocation method for cooperative manipulators utilizing the real-time task capability of the manipulators, unlike the wrench allocation method in [3, 4, 6].

The performance comparison of our proposed task capability finding algorithm with [9, 10] has been demonstrated in Section 5.1. To analyze the algorithm's efficacy for collaborative manipulators, the new task capability algorithm has been employed for task capability finding and task allocation in collaborative trajectory tracking tasks (Section 5.1).

The paper's organization is as follows. Section 2 describes a detailed literature survey. The background and problem definition of the cooperative manipulation are explained in Section 3, and the task capability algorithm and task allocation algorithm are depicted in Section 4. The result and conclusion are discussed in Sections 5 and 6.

2 Related Work

A robot's motion and wrench exertion capability are fundamentally significant for collaborative manipulation in workspace design and task coordination. Performance measures provide the required tool for quantifying these capabilities to apprise and optimize the characterization of manipulators. Specifically, the local indices [11] are vastly used in the application-level collaborative manipulation control and coordination design. The capability of a robotic mechanism was first quantified as a manipulability index by *Yoshikawa* [12] and then extended [13] to a dynamic manipulability which measures the volume of the manipulability ellipsoid. Even though the ellipsoid-based measures are well accepted because of their ease of computation, they often lead to underestimation of the capabilities. On the contrary, manipulability polytopes [8] accurately represent the motion and wrench capability in the task space, considering the joint space limits. A polytope-based vertex-search algorithm to compute the task-space force polytope's vertices has been introduced in [7, 8] using slack variables and singular value decomposition methods. The polytope-based capability computation method in [14, 15] has been formulated for task-oriented performance capability finding. The vertex-search algorithm for task-space polytope has further improved in [9, 10, 16] by reducing the computation complexity using geometric characteristics of the joint torque space. Though the algorithm notably reduces complexity and computation time, it does not allow real-time execution in cooperative manipulations.

Guay et al. [17] introduced the Capacity Margin (CM), a scalar polytope-based minimum degree of constraint satisfaction index that quantifies the robustness of the equilibrium of an object. The CM index determines a wrench-feasible workspace of cable-driven parallel robots [17, 18] with bounded forces. The index has been approximated to a continuous capacity margin function [19] using a hyperplane shifting method with neural network-based activation functions. These works quantify the capacity margin but do not compute the capacity, which is crucial for collaborative manipulators.

A measure of the task-space wrench and motion for a multi-manipulator system were specified [8, 20, 21] for collaborative manipulation of a rigid object considering predefined fixed task/load sharing in the task space and the joint space limits. In reality, the predefined task/load sharing often leads to inefficient utilization of the capabilities of individual manipulators. The coupled inverse dynamics of multiple cooperating manipulators have been introduced [22–24] for computing the dynamic load-carrying capacity (DLCC) in the joint space using D' Alembert's principle for two manipulator systems ignoring the inertia properties of the manipulated object. Analytical and optimization-based methods are presented in [25–30] for parallel manipulators compute force capabilities using scale factors. Due to their complexity and heavy computation, the above methods do not scale well for many manipulators and cannot work in online capability index computation.

Multiple manipulators collaboratively manipulate an object require a net wrench at its center of mass. The grasp matrix maps the applied wrench by the manipulators to the resultant wrench at the object's center of mass (CoM). The EE of the manipulators can exert a wrench on the object in numerous combination that results in the exact desired wrench in the object. Choosing the correct combination of wrenches by the EE poses a problem of input redundancy. The wrench/load allocation problem for cooperative manipulators manipulating an object has been addressed using a few different approaches like input wrench minimization problem [31–33], energy conservation-based method [5], quadratic optimization with linearity constraints [34] and

using pseudo-inverse of the grasp matrix [1, 35]. The pseudo-inverse approach may not provide [2] accurate input wrench resolutions. A physically motivated characterization of the wrench and an analytical expression for all non-squeezing load distributions has been formulated in *Erhart and Hirche* [3, 4]. Convex optimization-based scheme by *Donner et al.* [6] introduced a physically plausible wrench allocation using the applied wrench's internal and manipulation components as constraints.

The preceding discussion on prior work shows that the time-varying capability of individual manipulators has not been considered so far. Additionally, task allocation without a knowledge of the capability of each manipulator results in non-optimal cooperation. A significant tracking error and internal wrench arise when a cooperative task is assigned to an under-capable group of manipulators. Even for an over-capable group of manipulators, it is necessary to estimate the safety margin since an unexpected impact on the manipulated object may lead to large tracking errors. These do not appear to have been addressed in the existing literature.

3 Preliminaries

Consider a cooperative fixed base manipulation system of N mobile manipulators, and each manipulator has n_i no of joints ($i = 1, ..., N$). The cooperative manipulators grasp an object at its periphery, as shown in Fig. 1. $\{\boldsymbol{w}\}$ defines the world fixed reference frame, and a body coordinate frame $\{\boldsymbol{b}\}$ is attached to the object Center of Mass (CoM). Unless specifically mentioned, all the quantities are defined in the world frame $\{\boldsymbol{w}\}$. In this paper $\boldsymbol{I_a} \in \mathbb{R}^{a\times a}$ denotes $a \times a$ identity matrix, $\boldsymbol{0_{a\times b}} \in \mathbb{R}^{a\times b}$ represents a $a \times b$ null matrix. The following assumptions are made:

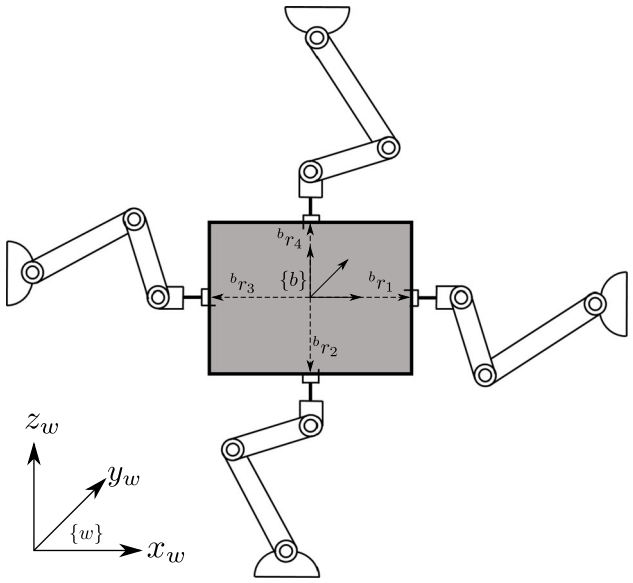

Fig. 1 Schematic of multi-robot cooperative object manipulation, relevant kinematic quantities are illustrated

Assumption 1 The grasped object is rigid.

Assumption 2 The manipulators grasp the object rigidly.

3.1 Wrench Capability of Manipulator

The generalized joint space coordinate of i-th manipulator of the cooperative manipulators having n_i joints is defined as $\boldsymbol{q_i} \in \mathbb{R}^{n_i}$. The EE's pose is defined in the Cartesian space by $\boldsymbol{x_i} = (\boldsymbol{p}_i^T, \boldsymbol{\theta}_i^T)^T \in \mathbb{R}^d$, where $\boldsymbol{p_i}, \boldsymbol{\theta_i}$ refers to the position and orientation ($d = 6$ in the 3D case and $d = 3$ in the 2D planar case). The task space velocities $\boldsymbol{v_i} \in \mathbb{R}^d$ of the EE is related to the joint space of the manipulator by the Jacobian matrix $\boldsymbol{J_i(q_i)} \in \mathbb{R}^{d\times n_i}$ utilizing the following Eq. 1.

$$\boldsymbol{v_i} = \boldsymbol{J_i(q_i)\dot{q}_i} \tag{1}$$

The task space force and torque applied by EE of i-th manipulator to the object at the grasp point is indicated by $\boldsymbol{f_i}, \boldsymbol{t_i} \in \mathbb{R}^3$ and concatenated to the *wrench* $\boldsymbol{h_i} = (\boldsymbol{f}_i^T, \boldsymbol{t}_i^T)^T$. The manipulator dynamics in the joint space expressed in Eq. 2 yields the mapping between the task space wrench $\boldsymbol{h_i} \in \mathbb{R}^d$ and the generalized joint torque $\boldsymbol{\tau_i} \in \mathbb{R}^{n_i}$.

$$\boldsymbol{M_i(q_i)\ddot{q}_i} + \boldsymbol{C_i(\dot{q}_i, q_i)\dot{q}_i} + \boldsymbol{g_i(q_i)} = \boldsymbol{\tau_i'} \tag{2a}$$

$$\boldsymbol{\tau_i'} + \boldsymbol{J}_i^T\boldsymbol{(q_i)h_i} = \boldsymbol{\tau_i} \tag{2b}$$

where $\boldsymbol{M_i(q_i)} \in \mathbb{R}^{n_i\times n_i}$ is the inertia matrix, $\boldsymbol{C_i(\dot{q}_i, q_i)} \in \mathbb{R}^{n_i\times n_i}$ represents the Coriolis and centrifugal matrix, $\boldsymbol{g_i(q_i)} \in \mathbb{R}^{n_i}$ is the effect of gravity. Eq. 2a provides the joint torque $\boldsymbol{\tau_i'}$ required for the manipulator's movement, and Eq. 2b computes the total joint torque requirement when the EE of the manipulator exerts an external wrench.

Equations 1 and 2 represent the kinematics and dynamics of a manipulator. The joint space velocities $\boldsymbol{\dot{q}_i}$ and torques $\boldsymbol{\tau_i}$ limits because of the physical limits of actuators and manipulator construction are defined as follows.

$$-\boldsymbol{\dot{q}}_{i,\max} \leq \boldsymbol{\dot{q}_i} \leq \boldsymbol{\dot{q}}_{i,\max} \tag{3a}$$

$$-\boldsymbol{\tau}_{i,\max} \leq \boldsymbol{\tau_i} \leq \boldsymbol{\tau}_{i,\max} \tag{3b}$$

Combining Eqs. 2b and 3b, the feasible wrench in the task space of the $i-th$ manipulator can be represented as a polytope using $2n_i$ linear inequalities.

$$-\boldsymbol{\tau}_{i,\max} - \boldsymbol{\tau_i'} \leq \boldsymbol{J}_i^T\boldsymbol{(q_i)h_i} \leq \boldsymbol{\tau}_{i,\max} - \boldsymbol{\tau_i'} \tag{4}$$

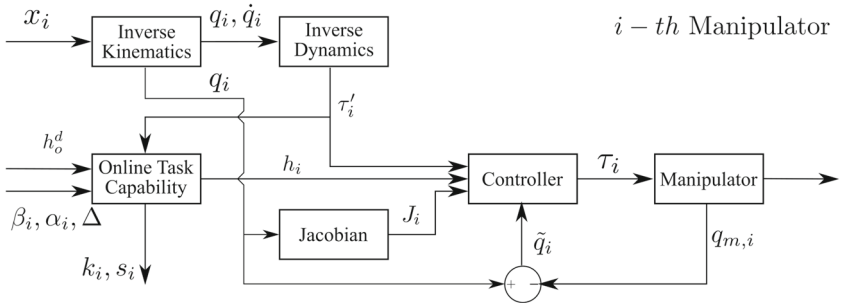

Fig. 2 Manipulator's task capability computation and control schematic

3.2 Cooperative Manipulation System Representation

The object CoM is located at $\boldsymbol{x_o} = [\boldsymbol{p}_o^T, \boldsymbol{\theta}_o^T]^T$. The EE grasping the object at ${}^{\boldsymbol{b}}\boldsymbol{r_i} \in \mathbb{R}^3$ defined in $\{\boldsymbol{b}\}$. The position of EE of i-th mobile manipulator can be express as $\boldsymbol{p_i} = \boldsymbol{p_o} + {}_{\boldsymbol{b}}^{\boldsymbol{w}}\boldsymbol{R}(\boldsymbol{\theta_o}){}^{\boldsymbol{b}}\boldsymbol{r_i}$, where ${}_{\boldsymbol{b}}^{\boldsymbol{w}}\boldsymbol{R}(\boldsymbol{\theta_o})$ is the rotation matrix of the body frame $\{b\}$ to the world frame $\{w\}$. Differentiating $\boldsymbol{p_i}$ in the world frame $\{\boldsymbol{w}\}$ gives $\dot{\boldsymbol{p}}_{\boldsymbol{i}} = \dot{\boldsymbol{p}}_{\boldsymbol{o}} + \boldsymbol{\omega_o} \times \boldsymbol{r_i}$, where $\boldsymbol{r_i} = {}_{\boldsymbol{b}}^{\boldsymbol{w}}\boldsymbol{R}(\boldsymbol{\theta_o}){}^{\boldsymbol{b}}\boldsymbol{r_i}$. The angular velocity of the object $\boldsymbol{\omega_o}$ and of the $i-th$ manipulator's EE $\boldsymbol{\omega_i}$ would be the same. The relationship between the i-th EE velocities $\boldsymbol{v_i} = (\dot{\boldsymbol{p}}_i^T, \boldsymbol{\omega}_i^T)^T \in \mathbb{R}^d$ and the velocities of the CoM of the object $\boldsymbol{v_o} = (\dot{\boldsymbol{p}}_o^T, \boldsymbol{\omega}_o^T)^T \in \mathbb{R}^6$ can be combined as following

$$\boldsymbol{v_i} = \underbrace{\begin{bmatrix} \boldsymbol{I_3} & -\boldsymbol{S}(\boldsymbol{r_i}) \\ \boldsymbol{0_{3\times3}} & \boldsymbol{I_3} \end{bmatrix}}_{\boldsymbol{G_i^T}} \boldsymbol{v_o} \tag{5}$$

where $\boldsymbol{S}(.)$ is the 3×3 skew-symmetric matrix operator that performs the cross product, $\boldsymbol{G_i} = \begin{bmatrix} \boldsymbol{I_3} & \boldsymbol{0_{3\times3}}; & \boldsymbol{S}(\boldsymbol{r_i}) & \boldsymbol{I_3} \end{bmatrix}$ represents the grasp matrix [36].

The equation of motion for the grasped object is expressed by

$$\boldsymbol{M_o}(\boldsymbol{x_o})\dot{\boldsymbol{v}}_{\boldsymbol{o}} + \boldsymbol{C_o}(\boldsymbol{x_o}, \dot{\boldsymbol{x}}_{\boldsymbol{o}})\boldsymbol{v_o} + \boldsymbol{g}(\boldsymbol{x_o}) = \boldsymbol{h}_o^d \tag{6}$$

where $\boldsymbol{M_o}(\boldsymbol{x_o}) \in \mathbb{R}^{d\times d}$ is the inertia matrix of the object, $\boldsymbol{C_o}(\boldsymbol{x_o}, \dot{\boldsymbol{x}}_{\boldsymbol{o}}) \in \mathbb{R}^{d\times d}$ is Coriolis and centrifugal matrix, $\boldsymbol{g}(\boldsymbol{x_o}) \in \mathbb{R}^d$ indicated gravitational force vector, $\boldsymbol{h}_o^d$ is the desired external wrench to be applied by EEs.

The resultant wrench acting on the object's CoM is denoted by $\boldsymbol{h_o} = (\boldsymbol{f}_o^T, \boldsymbol{t}_o^T)^T$ and is computed from the manipulators' wrenches utilizing the force and moment balance.

$$\boldsymbol{h_o} = \boldsymbol{G}[\boldsymbol{h_1^T}, \boldsymbol{h_2^T}, \cdots, \boldsymbol{h}_N]^T \tag{7}$$

Where $\boldsymbol{G} = [\boldsymbol{G_1}, \boldsymbol{G_2}, \cdots, \boldsymbol{G}_N] \in \mathbb{R}^{d\times dN}$ is the combined grasp matrix of the system.

The desired wrench $\boldsymbol{h}_o^d$ at the object Center of Mass is obtained using Eq. 6. Considering the dynamic task capabilities of the individual manipulators in the group, a new scheme for wrench distribution among the manipulators is proposed. The formulation of the wrench allocation scheme is described in Section 4.3.

4 Adaptive Task Allocation

In collaborative object manipulation, the task capability measure of individual manipulators and the collaborative group

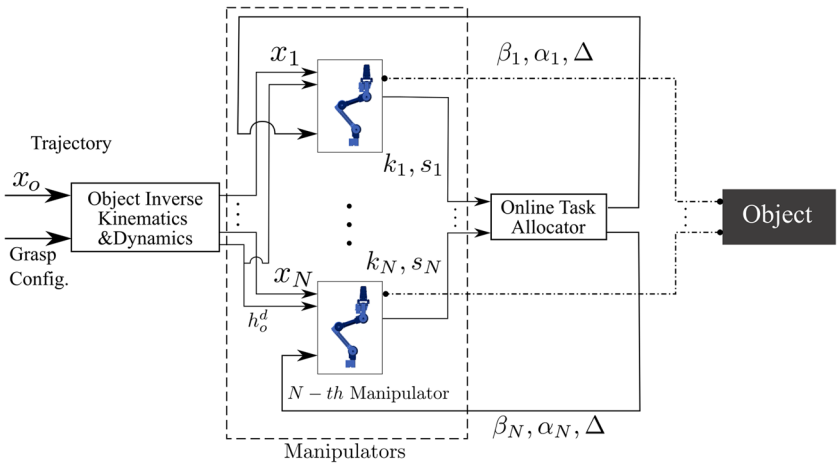

Fig. 3 Dynamic cooperation scheme

is necessary for successful task completion, task sharing, and computing the number of manipulators. Hence, we are introducing a novel online task capability finding Algorithm 2 and an adaptive cooperative task allocation Algorithm 3. Figure 2 indicates the task capability computation and control scheme of $i-th$ manipulator. The overall schematic of the dynamic cooperation scheme in collaborative object manipulation is shown in Fig. 3. A detailed description of the online task capability computation scheme and the adaptive task allocation schemes are presented in Sections 4.2 and 4.3.

4.1 Task Capability Finding Algorithm

The wrench capability varies with the task, and the manipulator's joint states $\boldsymbol{q_i}, \dot{\boldsymbol{q}}_{\boldsymbol{i}}$. We reformulate the task capability finding formulation in Eq. 4. The task space capability index of a manipulator is computed concerning the required wrench $\boldsymbol{h_o^d}$ for collaborative tasks utilizing a scalar factor k_i and the formulation is presented in Eq. 8.

$$-\boldsymbol{\tau_{i,\max}} - \boldsymbol{\tau_i'} \leq k_i \boldsymbol{J}_i^T(\boldsymbol{q_i})\boldsymbol{h_o^d} \leq \boldsymbol{\tau_{i,\max}} - \boldsymbol{\tau_i'} \tag{8}$$

Equation 6 computes the required wrench $\boldsymbol{h_o^d}$ to the object at its CoM for a cooperative trajectory-tracking task. The desired joint states $\boldsymbol{q_i}, \dot{\boldsymbol{q}}_{\boldsymbol{i}}$ of the manipulator are computed by solving Eq. 1. Equation 8 can be solved utilizing Linear Programming Problem (LPP) defined in Eq. 9.

$$\max_{k_i} \quad z = k_i \tag{9a}$$

$$\text{s.t.} \quad |\boldsymbol{\tau_i'} + k_i \boldsymbol{J}_i^T(\boldsymbol{q_i})\boldsymbol{h_o^d}| \leq \boldsymbol{\tau_{i,\max}} \tag{9b}$$

$$k_i \geq 0 \tag{9c}$$

Instead of solving the LPP, we solve the wrench capability finding problem in Eq. 9 with a novel approach that drastically reduces the complexity compared to the approach proposed by Skuric et al. in [10]. The proposed task capability computation method has been illustrated in Algorithm 1. The task capability finding for collaborative manipulators has been depicted in the following section.

Algorithm 1 Function: TaskCapability ($\boldsymbol{\tau_i'}, \boldsymbol{J}_i, \boldsymbol{h_o^d}$).

Input: $\boldsymbol{\tau_i'}, \boldsymbol{J}_i, \boldsymbol{h_o^d}$
Output: Task capabilities k_i

$\boldsymbol{a} = \boldsymbol{J}_i^T \boldsymbol{h_o^d}$
$\boldsymbol{b_{min}} = -\boldsymbol{\tau}_{i,max} - \boldsymbol{\tau_i'}$
$\boldsymbol{b_{max}} = \boldsymbol{\tau}_{i,max} - \boldsymbol{\tau_i'}$
$\boldsymbol{x}^k = max(\boldsymbol{b}^k_{min}/\boldsymbol{a}^k, \boldsymbol{b}^k_{max}/\boldsymbol{a}^k), \forall k = 1, .., n_i$
$k_i = min(\boldsymbol{x})$
return k_i

4.2 Online Task Capability of Cooperative Manipulators

Algorithm 1 finds the task capability indexes $\boldsymbol{k} = (k_1, k_2, \cdots, k_N)^T$ and the total task capacity index of a cooperative manipulator group is computed as a scalar $X_1 = \sum_{i=1}^{N} k_i$, where X_1 is the scaling parameter of the total task ($\boldsymbol{h_o^d}$) i.e. the maximum functional wrench capability of the cooperative groups would be $X_1 * \boldsymbol{h_o^d}$. The EE of the manipulators applies force at grasp points on the periphery of the object, which results in undesired moments about the CoM of the object. To counterbalance that, it requires an equal and opposite moment $\boldsymbol{t_\Delta}$ computed from Eq. 10.

$$\begin{aligned} \boldsymbol{t_\Delta} &= \sum_{i=1}^{N} \boldsymbol{r_i} \times \beta_i \boldsymbol{f_o} \\ &= \boldsymbol{\Delta} \times \boldsymbol{f_o} \end{aligned} \tag{10}$$

where a scalar quantity β_i indicates the dynamic wrench allocation coefficient of $i-th$ manipulator, $\boldsymbol{\Delta} = \sum_{i=1}^{N} \beta_i \boldsymbol{r_i}$. The cooperative manipulator groups must produce an additional moment to compensate $\boldsymbol{t_\Delta}$. Application of torque by EEs is prioritized over force for producing $\boldsymbol{t_\Delta}$. Only some manipulators can contribute towards the disturbance moment on top of the maximum functional task capability [25, 37]. Our proposed measure also considers a scheme for the task capability contribution from the disturbance moment since these moments help the manipulator enhance its task capability. The proposed scheme is an optimization exercise similar to Eq. 9.

$$\begin{aligned} \max_{s_i} \quad & z = s_i \\ \text{s.t.} \quad & |\boldsymbol{\tau_i'} + k_i \boldsymbol{J}_i^T(\boldsymbol{q_i})\boldsymbol{h_o^d} + s_i \boldsymbol{J}_i^T(\boldsymbol{q_i})\boldsymbol{h_\Delta}| \leq \boldsymbol{\tau_{i,\max}} \\ & s_i \geq 0 \end{aligned} \tag{11}$$

where $\boldsymbol{h_\Delta} = \begin{Bmatrix} \boldsymbol{0_3} \\ -\boldsymbol{t_\Delta} \end{Bmatrix}$ and s_i is the measure of additional capability towards $\boldsymbol{t_\Delta}$. To completely produce $\boldsymbol{h_\Delta}$, the total capability ($\sum_{i=1}^{N} s_i \geq 1$) should be ≥ 1. If ($\sum_{1}^{N} s_i < 1$), then the deficiency in counterbalance moment needs to be settled at the cost of task capability X_1. This additional capability $\boldsymbol{s} = (s_1, s_2, \cdots, s_N)^T$ have the potential to improve [25, 37] the task capability $\boldsymbol{k}$ indexes. To incorporate this effect into $\boldsymbol{k}$ the renewed contribution in h_Δ can be measured by modifying the Eq. 9 as follows

$$\begin{aligned} \max_{k_i} \quad & z = k_i \\ \text{s.t.} \quad & |\boldsymbol{\tau_i'} + k_i \boldsymbol{J}_i^T(\boldsymbol{q_i})\boldsymbol{h_o^d} + \alpha_i \boldsymbol{J}_i^T(\boldsymbol{q_i})\boldsymbol{h_\Delta}| \leq \boldsymbol{\tau_{i,\max}} \\ & k_i \geq 0 \end{aligned} \tag{12}$$

But in reality, the Eqs. 11 and 12 are interlinked implicitly, which is complex to solve in real-time implementation. Equation 12 is further converted to an explicit relation using the value from the previous time step.

$$\begin{aligned} \max_{k_i} \quad & z = k_i \\ \text{s.t.} \quad & |\boldsymbol{\tau_i'} + k_i \boldsymbol{J}_i^T(\boldsymbol{q_i})\boldsymbol{h_o^d} + \alpha_i' \boldsymbol{J}_i^T(\boldsymbol{q_i})\boldsymbol{h'_\Delta}| \leq \boldsymbol{\tau_{i,\max}} \\ & k_i \geq 0 \end{aligned} \tag{13}$$

where α_i' are taken from the previous time step t' and $\boldsymbol{h'_\Delta}$ is calculated by substituting β_i' in place of β_i in the Eq. 10.

The task capability computation for $i-th$ manipulators of the cooperative manipulators is presented in Algorithm 2. Figure 2 shows $i-th$ manipulator's online task capability computation and control schematic.

Algorithm 2 Online task capability.

Input: $\alpha_i', \boldsymbol{\tau_i'}, \boldsymbol{J}_i, \boldsymbol{h_o^d}, \boldsymbol{h'_\Delta}$
Output: Task capabilities k_i, s_i

$\boldsymbol{\tau'_{i,\Delta}} = \boldsymbol{\tau_i'} + \alpha_i' \boldsymbol{J}_i^T \boldsymbol{h'_\Delta}$
k_i = TaskCapability($\boldsymbol{\tau'_{i,\Delta}}, \boldsymbol{J}_i, \boldsymbol{h_o^d}$)
$\boldsymbol{\tau'_{i,o}} = \boldsymbol{\tau_i'} + k_i \boldsymbol{J}_i^T \boldsymbol{h_o^d}$
s_i = TaskCapability($\boldsymbol{\tau'_{i,o}}, \boldsymbol{J}_i, \boldsymbol{h_o^d}$)
return k_i, s_i

4.3 Task Allocation

The individual manipulators share the collaborative task proportional to their task capability $\boldsymbol{k}$ obtained from Algorithm 2 for efficiently utilizing the manipulators. Dynamic task allocation coefficient $\boldsymbol{\beta} = (\beta_1, \beta_2, \cdots, \beta_N)^T$ is obtained from the total task capability using the following Eq. 14.

$$\boldsymbol{\beta} = \frac{\boldsymbol{k}}{\sum_{i=1}^{N} k_i} \tag{14}$$

where the task allocation co-efficient β_i has the following properties

$$\sum_{i=0}^{N} \beta_i = 1, \ and \ \beta_i \in [0,1] \ \forall i = 1, \ldots, N \tag{15}$$

The wrench contribution of the $i-th$ manipulator toward the task would be as follows.

$$\boldsymbol{h_{o,i}} = \beta_i \boldsymbol{h_o^d} \tag{16}$$

The contribution of i-th manipulator $\boldsymbol{h_{o,i}}$ would be within its maximum capability $k_i \boldsymbol{h_o^d}$ at any time instant

$$\boldsymbol{h_{o,i}} \leq k_i \boldsymbol{h_o} \tag{17}$$

After substituting Eq. 16 into Eq. 17 and comparing it with Eq. 14 it is obtained that for successful task completion, the value of total task capability, $X_1(\sum_1^N k_i)$ should always be more than or equal to one.

The individual manipulators contributes the additional required wrench $\boldsymbol{h_\Delta}$ in proportion to the capability measure s_i obtained from Eq. 11 in the following manner:

- When the total capability $\sum s_i$ is ≥ 1 the h_Δ could be entirely allocate as follows

$$\boldsymbol{\alpha} = \frac{\boldsymbol{s}}{\sum_{i=1}^{N} s_i} \tag{18}$$

- When the total capability $\sum s_i$ is ≤ 1 the h_Δ could be allocate as follows

$$\alpha_i = \boldsymbol{s} + (1 - \sum_{i=1}^{N} s_i) * \boldsymbol{\beta} \tag{19}$$

where, $\alpha_i \in \mathbb{R}$ with $0 \leq \alpha_i \leq 1$ represents the i-th manipulator's share of additional wrench requirement to counterbalance h_Δ.

$$\sum_{i=1}^{N} \alpha_i = 1 \ and \ \alpha_i \in [0,1] \ \forall i = 1, \ldots, N \tag{20}$$

The collaborative task share of $i-th$ manipulator for the disturbance wrench is obtained using Eq. 21.

$$\boldsymbol{h_{\Delta,i}} = \alpha_i \boldsymbol{h_\Delta} \tag{21}$$

The total wrench contribution(h_i) of i-th manipulator consists of two components: one is the required object wrench for the task completion, and another is for compensating the disturbance moment $\boldsymbol{t_\Delta}$.

$$\boldsymbol{h_i} = \boldsymbol{h_{o,i}} + \boldsymbol{h_{\Delta,i}} = \beta_i \boldsymbol{h_o} + \alpha_i \boldsymbol{h_\Delta} \tag{22}$$

The unified online task allocation algorithm has been presented in Algorithm 3. Figure 3 shows the working of the online task allocation in the collaborative object manipulation.

Algorithm 3 Online task allocator.

Input: Task capabilities $\boldsymbol{k}, \boldsymbol{s}$
Output: $\Delta, \boldsymbol{\beta}, \boldsymbol{\alpha}$

$X_1 = \sum_1^N \boldsymbol{k}_i$
$X_2 = \sum_1^N \boldsymbol{s}_i$
if $X_1 \leq 1$ **then**
  $\boldsymbol{\beta} = \boldsymbol{k}$
else
  $\boldsymbol{\beta} = \boldsymbol{k}/X_1$
end if
$\boldsymbol{\Delta} = \sum_{i=1}^N \beta_i \boldsymbol{r}_i$
if $X_1 \leq 1$ **then**
  $\boldsymbol{\alpha} = \boldsymbol{s} + (1 - X_2) * \boldsymbol{\beta}$
else
  $\boldsymbol{\alpha} = \boldsymbol{s}/X_2$
end if

4.4 Control for Individual Robot

For the desired trajectory tracking of the object, the passivity-based controller [38, 39] for i-th robot has been designed as the following,

$$\boldsymbol{\tau_i} = \boldsymbol{\tau_i'} + \boldsymbol{J}_i^T(\boldsymbol{q_i})\boldsymbol{h_i} + \boldsymbol{K_p}\tilde{\boldsymbol{q}}_i + \boldsymbol{K_v}\dot{\tilde{\boldsymbol{q}}}_i \tag{23}$$

where $\tilde{\boldsymbol{q}}_i, \dot{\tilde{\boldsymbol{q}}}_i$ are the error vectors of joint angle, joint angular velocity, and $\boldsymbol{K_p}, \boldsymbol{K_v}$ are the respective gain matrix.

5 Results

The proposed online task capability finding algorithm in Section 4.1 has been evaluated for complexity. Section 5.1 reports the comparison with the existing state-of-art algorithms. To validate the efficiency of the proposed online task capability computation and task allocation algorithm illustrated in Sections 4.2 and 4.3 respectively, collaborative manipulators are employed for object transportation in MATLAB Simulink and Simscape simulation.

5.1 Task Capability Algorithm Efficacy Analysis

The efficiency of the proposed task capability algorithm has been demonstrated and compared with the algorithm proposed by Skurik et al. [10] and is further broadened to the algorithm proposed by Sasaki et al. [9]. We employ all three algorithms to compute the task capability for two industrial manipulators: the Universal Robots UR5 6 DoF robot and the Franka Emika Panda 7 DoF robot. The test was done in the MATLAB script on 5000 randomly generated configurations and tasks.

Table 1 shows the computation time evaluation. Our proposed algorithm is 12× faster than Skuric's algorithm and 9× faster than Sasaki's. The performance evaluation shows that for the case of industrial 6 DoF and 7 DoF manipulators, our methodology can compute the task capability under $15\mu - sec$. Such faster computation extends the online use of capability evaluation in collaborative task allocation.

Table 1 Computation time (in microseconds) comparison with the state of art algorithm

Robot	Time ($\mu - sec$)	Skuric [10]	Sasaki [9]	Proposed
UR5	min	107.6	76.10	8.50
	mean	185.9	94.43	11.81
	max	515.7	571.7	72.80
	sd	48.64	19.51	3.04
Panda	min	118.1	83.10	8.60
	mean	214.0	101.53	11.76
	max	740.4	594.8	86.80
	sd	50.28	17.94	3.07

5.2 Simulation Setup

Three separate trajectories with multiple manipulators varying between three and four are evaluated to demonstrate the efficiency of the proposed capability finding and task allocation algorithm. Multiple OpenMANIPULATOR-X [40] shown in Fig. 4 with the joint 1 locked are involved in this simulation. All EEs rigidly grasp the rectangular plate-like object on its periphery. The grasp configuration and the base mounting position of the manipulator are mentioned in Table 2. The object's motion is allowed in the vertical X-Z plane with the gravity constant $g = 9.8067 m/sec^2$ acting in the negative Z direction. The joint actuators' torque limit is set to $\pm 1 Nm$. A detailed manipulator description is available on the [40].

The object's description is specified in Table 3. The EE motions are coordinated to follow a desired trajectory by the object's center of mass (CoM) utilizing the dynamic cooperation scheme shown in Fig. 3. The simulations have been

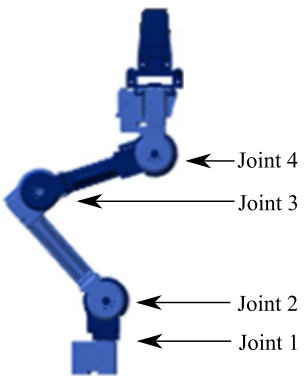

Fig. 4 Schematic of the OpenMANIPULATORX [40]

Table 2 Position of the manipulator base and EE grasping position

Config.	Joint 2 Position (m)	EE Position in frame $\{b\}$
A	$\boldsymbol{O_1} = [0.7, 0, 0.35]^T$	${}^b\boldsymbol{r}_1 = [0.1, 0, 0]^T$
	$\boldsymbol{O_2} = [0.35, 0, 0]^T$	${}^b\boldsymbol{r_2} = [0, 0, -0.075]^T$
	$\boldsymbol{O_3} = [0, 0, 0.35]^T$	${}^b\boldsymbol{r_3} = [-0.1, 0, 0]^T$
	$\boldsymbol{O_4} = [0.35, 0, 0.7]^T$	${}^b\boldsymbol{r_4} = [0, 0, 0.075]^T$
B	$\boldsymbol{O_1} = [0.43, 0, 0]^T$	${}^b\boldsymbol{r_1} = [0.08, 0, -0.075]^T$
	$\boldsymbol{O_2} = [0.27, 0, 0]^T$	${}^b\boldsymbol{r_2} = [-0.08, 0, -0.075]^T$
	$\boldsymbol{O_3} = [0.27, 0, 0.7]^T$	${}^b\boldsymbol{r_3} = [-0.08, 0, 0.075]^T$
	$\boldsymbol{O_4} = [0.43, 0, 0.7]^T$	${}^b\boldsymbol{r_4} = [0.08, 0, 0.075]^T$
C	$\boldsymbol{O_1} = [0.43, 0, 0]^T$	${}^b\boldsymbol{r_1} = [0.08, 0, -0.075]^T$
	$\boldsymbol{O_3} = [0.27, 0, 0.7]^T$	${}^b\boldsymbol{r_3} = [-0.08, 0, 0.075]^T$
	$\boldsymbol{O_4} = [0.43, 0, 0.7]^T$	${}^b\boldsymbol{r_4} = [0.08, 0, 0.075]^T$

performed on three different trajectories, mentioned in the following:

T1: Circular: $\boldsymbol{x_o} = [0.35 + 0.05\cos(0.4\pi t), 0, 0.35 + 0.05\sin(0.4\pi t)]^T m$,
T2: Infinity: $\boldsymbol{x_o} = [0.35 + 0.05\cos(0.4\pi t), 0, 0.35 + 0.025\sin(0.8\pi t)]^T m$,
T3: Diverging Sinusoidal: $\boldsymbol{x_o} = [0.3 + 0.01t + 0.001t^3, 0, 0.35 + 0.01t\cos(2\pi t)]^T m$.

Three different collision-free configurations (Config.) A, B, and C have been chosen for the simulation. The grasping points are distributed along the periphery of the object to minimize the disturbing moment. However, other possible configurations satisfy these conditions. Schematic diagrams of the three different Config. are presented in Fig. 5. All the manipulators are assumed to be identical. The orientation of the EE's is perpendicular to the object's edge.

Config. A **:** Four manipulators are grasping the object from four sides, left-right and top-bottom.
Config. B **:** Four manipulators are grasping the object from two sides, two from the top and two from the bottom.
Config. C **:** Three manipulators are grasping the object from two sides, two from the top and one from the bottom.

5.3 Trajectory Tracking

The desired trajectory $\boldsymbol{x_o}$ of object CoM is shared with the individual manipulators. The desired wrench ($\boldsymbol{h_o^d}$), which needs to be applied to the object by the EEs, is calculated using Eq. 6 by the manipulators.

5.4 Circular Trajectory

Figure 6 plots the individual and total task capability of individual manipulators and the total task capability of the group for Config. A, B, and C for 20 seconds while the object CoM follows the circular trajectory **T1**. This total capability measure should be more than one at every instant for successful trajectory tracking. For the Config. Figure 6(a) shows that the minimum total capability of the group is 0.90, which is less than one. It indicates a significant deviation from the desired trajectory if this group participates in the task. Config. B is the result of the reorientation of the least competent manipulators 1 and 3 of Config. A. The least capable manipulators are identified from Fig. 6(a). The alternative is the use of additional manipulators.

The task capability of Config. B presented in Fig. 6(b) indicates the minimum value of the total task capability that improves to 1.47 with reoriented manipulators, i.e., an enhancement of 63.3%. The capability is well above unity, indicating that the trajectory tracking error will be much lower for Config. B than for Config. A. The extensive capability margin of 0.47 above 1 in Config. B initiates the scope of reducing the number of manipulators if the minimum total task capability remains above one after reduction. Therefore, in Config. C, manipulator 2 is removed, and manipulator 1 is translated. The task capability plot for Config. C is presented in Fig. 6(c). The minimum value of total task capability is 1.00, and there is no safety margin available for an unforeseen disturbance. Config. C can be employed for the task that accepts performance deviation from the desired trajectory due to an unforeseen disturbance.

Since the goal of collaborative manipulation here is trajectory tracking, the tracking error defines the performance. The trajectory tracking performances for Config. A, B, and C are presented in Fig. 7. Figure 7(a,c,e) shows the deviation of the object from the desired trajectory, while Fig. 7(b,d,e) quantifies trajectory tracking error. While analyzing these, we ignore the initial transient phase error and look at the steady-state error. The maximum steady-state errors in the Z direction for Config. A, B, and C are $-2.27 \times 10^{-3} m$,

Table 3 Parameter of the grasped object

Mass (kg)	Inertia ($\times 10^-3 kg.m^2$)	Dimensions (m)	CoM
2.4	$diag([3.8, 10.4, 6.7])$	$[l_x, l_y, l_z] = [0.2, 0.02, 0.15]$	Geometric Midpoint

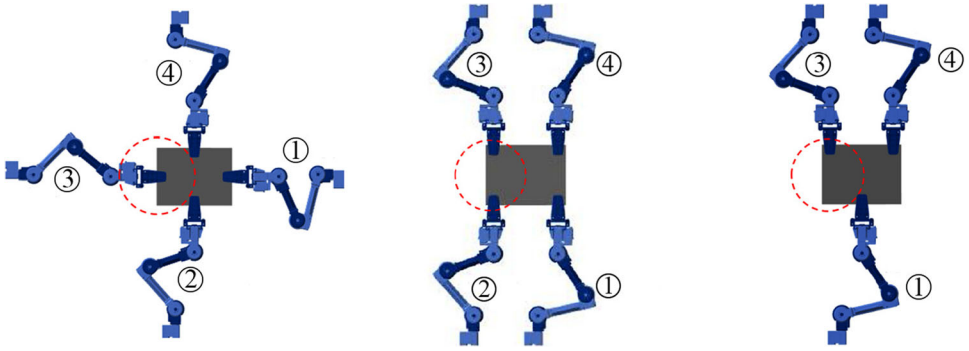

Fig. 5 Multi-manipulator team manipulates the object in three configurations. The object's trajectory is described by the dashed circle for T1

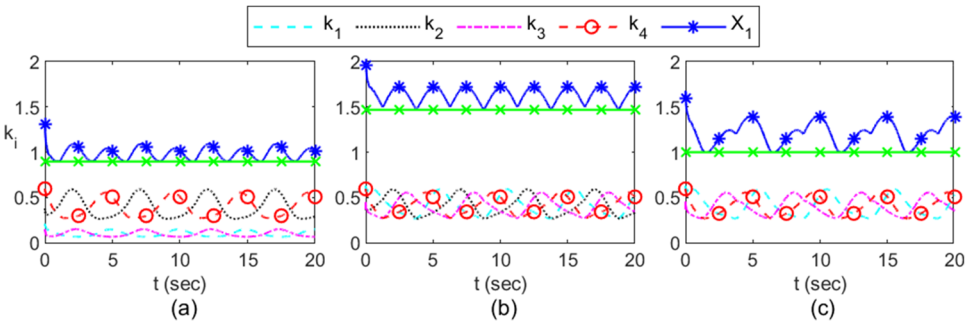

Fig. 6 Task Capability of individual manipulators and total task capacity of the group for the three different configurations (a) Config. A, (b) Config. B and (c) Config. C for trajectory **T1**. The green straight line indicates the minimum of the total task capability

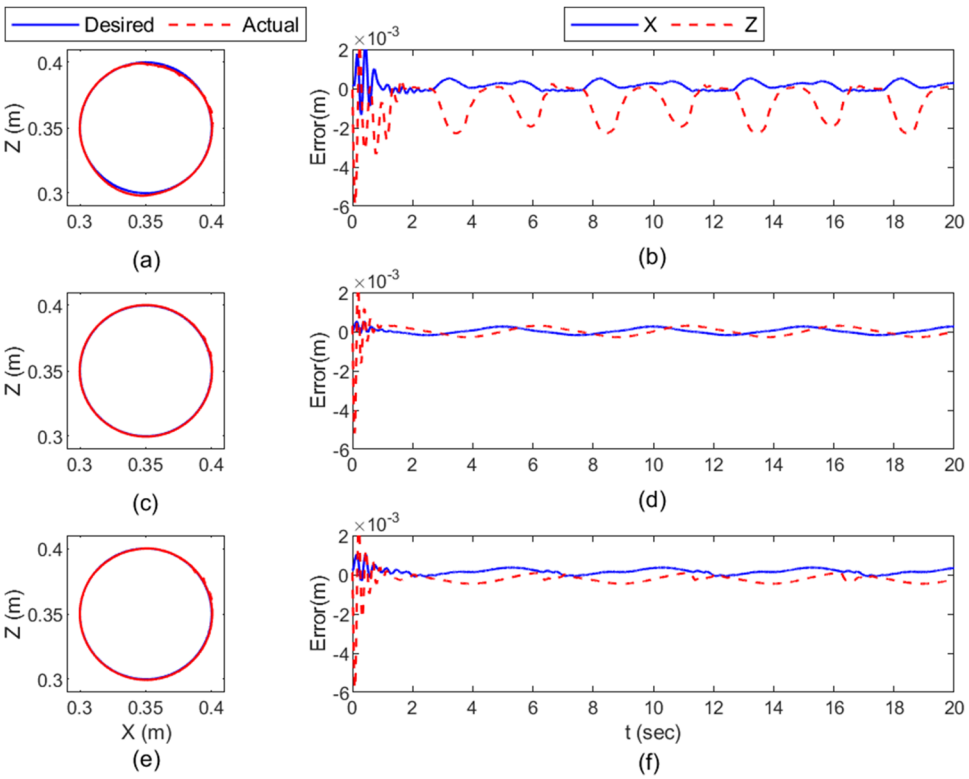

Fig. 7 The trajectory tracking performance for Config. A, B, and C are presented. (a), (c) and (e) represents the comparison of desired and actual trajectory tracked, and (b), (d), and (f) represent the tracking errors for trajectory **T2** of Config. A, B, and C, respectively

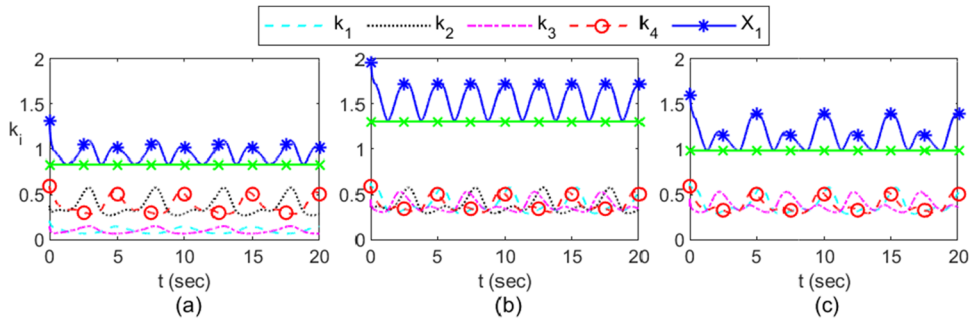

Fig. 8 Task Capability of individual manipulators and total task capacity of the group for the three different (a) Config. A, (b) Config. B and (c) Config. C. The green straight line indicates the minimum of the total task capability

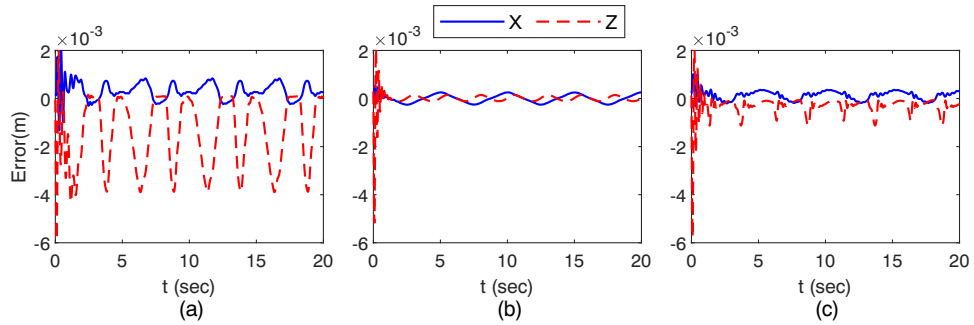

Fig. 9 The trajectory tracking performance for Config. A, B, and C in infinity-like trajectory **T2** shown in (a),(b), and (c), respectively

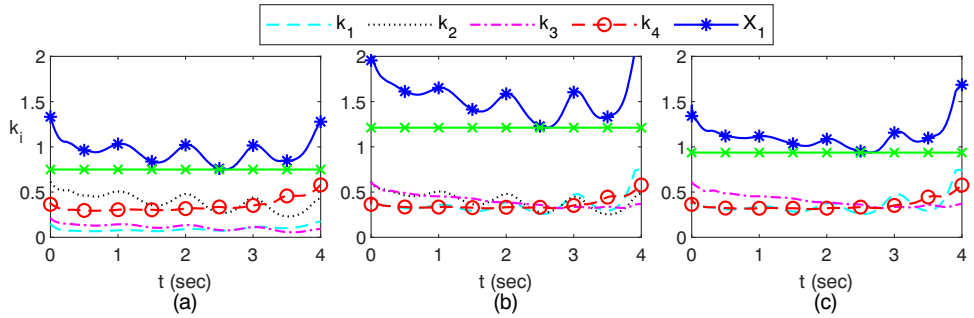

Fig. 10 Task Capability of individual manipulators and total task capacity of the group (a) Config. A, (b) Config B and (c) Config. C. The green straight line indicates the minimum of the total task capability

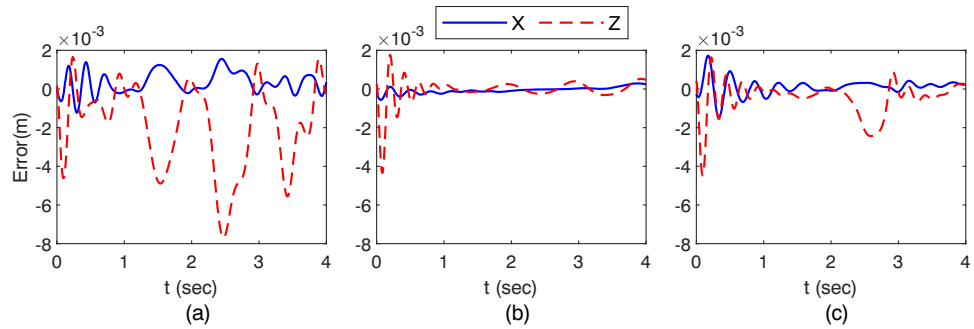

Fig. 11 The trajectory tracking errors of sinusoidal trajectory **T3** for Config. A, B, and C are shown in (a),(b), and (c), respectively

$\pm 0.299 \times 10^{-3}m$ and $\pm 0.461 \times 10^{-3}m$ respectively at the instant when the total capability is at minimum value. While the total task capability of Config. A drops below 1.0, the error starts growing, which matches our expectations based on the task capabilities of the group of manipulators for the three configurations.

5.5 Infinity Trajectory

The simulation presented in Section 5.4 is repeated for trajectory **T2**. Figure 8(a),(b), and (c) shows the individual and total task capability of Config. A, B, and C, respectively. The minimum values of total task capability are 0.83, 1.3, and 0.98, . The Config. C, which is just feasible for a circular trajectory **T1**, is infeasible for an infinity-shaped trajectory **T2**. Figure 9 shows the corresponding trajectory tracking error. The maximum steady-state errors in Config. A, B and C are $-3.86 \times 10^{-3}m$, $\pm 0.256 \times 10^{-3}m$ and $-1.118 \times 10^{-3}m$ respectively. The deterioration in trajectory tracking error from $\pm 0.461 \times 10^{-3}m$ to $-1.118 \times 10^{-3}m$ for Config. C, when the trajectory changes from **T1** to **T2** and the metric changes from 1.0 to 0.98, reaffirms our confidence in using the proposed task capability metric.

5.6 Diverging Sinusoidal Trajectory

The simulations for diverging sinusoidal trajectory **T3** are executed for 4 seconds. The minimum value of total task capability for Config. A, B, and C are 0.75, 1.12, and 0.94, shown in Fig. 10. Hence, among all, only Config. B is expected to perform well. This is indeed borne out by the plots of trajectory and tracking error shown in Fig. 11. The maximum value of the steady-state trajectory tracking error is $-7.69 \times 10^{-3}m$, $\pm 0.398 \times 10^{-3}m$ and $-2.43 \times 10^{-3}m$ for Config. A, B, and C, respectively.

The total task capability of the cooperative manipulators should always be greater than unity for efficient trajectory tracking. For Config. B, this metric changed from 1.47 to 1.3 and finally to 1.12 as the trajectory changed from **T1** to **T2** and to **T3**, however, for Config. C, the corresponding values were 1.0, 0.98, and 0.94. The smaller range of variation for Config. C compared to Config. B that does not alter the basic concept behind the proposed metric, i.e., any value lower than unity would significantly deteriorate trajectory tracking error.

6 Conclusion

The proposed task capability finding algorithm reduces the computation time for capability finding of a manipulator by 85% than the existing state-of-art algorithm that boosts its online use in collaborative cases. The proposed algorithm enables online adaptive task allocation proportional to the individual manipulator's capability. However, a different dynamic allocation scheme may be explored based on other criteria like minimum disturbance wrench. The trajectory tracking performance for the three trajectories in three manipulator configurations shows the efficacy of the proposed metric. The effect of the change of trajectory and the configuration of the manipulators has been studied. The trend of the trajectory tracking errors for these nine cases matches the predictions of the proposed metric. The proposed method also aids in identifying the least capable manipulator among a group of manipulators and assists in the efficient utilization of the capability of the group. Task capacity margin should be considered for more efficient performance with the uncertainty in the required wrench. One can also examine the effect of different base and EE grasping positions while ensuring that there would be no collision and kinematic discrepancies among the members. The proposed capability finding algorithm may not work well along with the optimization-based motion planning of collaborative manipulation because of the use of the max-min function in the algorithm. Our future works would address the shortcoming.

Author Contributions K. Patra conceived the idea, developed the theoretical framework, and performed simulations. A. Sinha devised the simulation test cases. A. Guha contributed to interpreting the results.

Availability of data and materials The datasets generated during the current study are available from the corresponding author on specific request.

References

1. Bonitz, R.G., Hsia, T.C.: In: Proceedings of the 1994 IEEE International Conference on Robotics and Automation, vol. 2, pp. 1521–1527 (1994). https://doi.org/10.1109/ROBOT.1994.351372
2. Chung, J.H., Yi, B.J., Kim W.K.: Analysis of internal loading at multiple robotic systems. J. Mech. Sci. Technol **19** (2005). https://doi.org/10.1007/BF03023933
3. Erhart, S., Hirche, S.: Internal force analysis and load distribution for cooperative multi-robot manipulation. IEEE Trans. Robot. **31**(5), 1238–1243 (2015). https://doi.org/10.1109/TRO.2015.2459412

4. Erhart, S., Hirche, S.: Model and analysis of the interaction dynamics in cooperative manipulation tasks. IEEE Trans. Robot. **32**(3), 672–683 (2016). https://doi.org/10.1109/TRO.2016.2559500
5. Schmidts, A.M., Schneider, M., Kühne, A.: Peer, In: 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 4922–4929 (2016). https://doi.org/10.1109/ICRA.2016.7487698
6. Donner, P., Endo, S., Buss, M.: Physically plausible wrench decomposition for multieffector object manipulation. IEEE Trans. Robot. **34**(4), 1053–1067 (2018). https://doi.org/10.1109/TRO.2018.2830369
7. Chiacchio, P., Bouffard-Vercelli, Y., Pierrot, F.: In: Proceedings of IEEE International Conference on Robotics and Automation, vol. 4 pp. 3520–3525 (1996). https://doi.org/10.1109/ROBOT.1996.509249
8. Chiacchio, P., Bouffard-Vercelli, Y., Pierrot, F.: Force polytope and force ellipsoid for redundant manipulators. J. Robot. Syst. **14**(8), 613–620 (1997). https://doi.org/10.1002/(SICI)1097-4563(199708)14:8<613::AID-ROB3>3.0.CO;2-P
9. Sasaki, M., Iwami, T., Miyawaki, K., Sato, I., Obinata, G., Dutta A.: In: Search Algorithms and Applications, (ed.) by Mansour, N., (IntechOpen, Rijeka, 2011), chap. 22. https://doi.org/10.5772/14201
10. Skuric, A., Padois, V., Daney, D.: In: 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 1700–1706 (2021). https://doi.org/10.1109/ICRA48506.2021.9562050
11. Patel, S., Sobh, T.: Manipulator performance measures – a comprehensive literature survey. J Intell. Robot. Syst. Theory Appl. **77**(3-4), 547–570 (2015). https://doi.org/10.1007/s10846-014-0024-y. https://www.scopus.com/inward/record.uri?eid=2-s2.0-84924223466&doi=10.1007%2fs10846-014-0024-y&partnerID=40&md5=4c5058798c844682fae75ad123e83515. Cited by: 166; All Open Access, Green Open Access
12. Yoshikawa, T.: Manipulability of robotic mechanisms. Int. J. Robot. Res. **4**, 3–9 (1985). https://doi.org/10.1177/027836498500400201
13. Yoshikawa, T.: in Proceedings. IEEE International Conference on Robotics and Automation **2**(1985), 1033–1038 (1985). https://doi.org/10.1109/ROBOT.1985.1087277
14. Stradovnik, S., Hace, A.: Task-oriented evaluation of the feasible kinematic directional capabilities for robot machining. Sensors **22**(11) (2022). https://doi.org/10.3390/s22114267. https://www.mdpi.com/1424-8220/22/11/4267
15. Feller, D.: Dexterity, workspace and performance analysis of the conceptual design of a novel three-legged, redundant, lightweight, compliant, serial-parallel robot. J. Intell. & Robot. Syst. **109**, 6 (2023). https://doi.org/10.1007/s10846-023-01900-8
16. Skuric, A., Padois, V., Rezzoug, N., Daney, D.: On-line feasible wrench polytope evaluation based on human musculoskeletal models: an iterative convex hull method. IEEE Robotics and Automation Letters **7**(2), 5206–5213 (2022). https://doi.org/10.1109/LRA.2022.3155374
17. Guay, F., Cardou, P., Cruz-Ruiz, A.L., Caro, S.: in New Advances. In: Petuya, V., Pinto, C., Lovasz, E.C. (eds.) Mechanisms, Transmissions and Applications, pp. 385–392. Springer, Netherlands, Dordrecht (2014)
18. Rasheed, T., Long, P., Marquez-Gamez, D., Caro, S.: In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 962–967 (2018). https://doi.org/10.1109/ICRA.2018.8461199
19. Sagar, K., Caro, S., Padır, T.: Long, P.: Polytope-based continuous scalar performance measure with analytical gradient for effective robot manipulation. IEEE Robot. Autom. Lett. **8**(11), 7289–7296 (2023). https://doi.org/10.1109/LRA.2023.3313926
20. Chiacchio, P., Chiaverini, S., Sciavicco, L., Siciliano, B.: Global task space manipulability ellipsoids for multiple-arm systems. IEEE Trans. Robot. Autom. **7**, 678–685 (1991). https://doi.org/10.1109/70.97880
21. Chiacchio, P., Chiaverini, S., Sciavicco, L., Siciliano, B.: Task space dynamic analysis of multiarm system configurations. Int. J. Robot. Res. **10**(6), 708–715 (1991). https://doi.org/10.1177/027836499101000608
22. Wang, L.C., Kuo, M.J.: Dynamic load-carrying capacity and inverse dynamics of multiple cooperating robotic manipulators. IEEE Trans. Robot. Autom. **10**(1), 71–77 (1994). https://doi.org/10.1109/70.285588
23. Sheng Zhao, Y., Lu, L., Shi Zhao, T., Hui Du, Y., Huang, Z.: The novel approaches for computing the dynamic load-carrying capacity of multiple cooperating robotic manipulators. Mech. Mach. Theory **34**(4), 637–643 (1999). https://doi.org/10.1016/S0094-114X(97)00107-9. https://www.sciencedirect.com/science/article/pii/S0094114X97001079
24. Zhao, Y.S., Ren, J.Y., Huang, Z.: Dynamic loads coordination for multiple cooperating robot manipulators. Mech. Mach. Theory **35**(7), 985–995 (2000). https://doi.org/10.1016/S0094-114X(99)00052-X. https://www.sciencedirect.com/science/article/pii/S0094114X9900052X
25. Nokleby, S., Fisher, R., Podhorodeski, R., Firmani, F.: Force capabilities of redundantly-actuated parallel manipulators. Mech. Mach. Theory **40**, 578–599 (2005). https://doi.org/10.1016/j.mechmachtheory.2004.10.005
26. S. Nokleby, F. Firmani, A. Zibil, R. Podhorodeski, Force-moment capabilities of redundantly-actuated planar-parallel architectures. Proceedings of the IFToMM pp. 17–21(2007)
27. Firmani, F., Zibil, A., Nokleby, S.B., Podhorodeski, R.P.: Wrench capabilities of planar parallel manipulators. part i: Wrench polytopes and performance indices. Robotica **26**(6), 791–802 (2008). https://doi.org/10.1017/S0263574708004384
28. Firmani, F., Zibil, A., Nokleby, S.B., Podhorodeski, R.P.: Wrench capabilities of planar parallel manipulators. part i: Wrench polytopes and performance indices. Robotica **26**(6), 791–802 (2008). https://doi.org/10.1017/S0263574708004384
29. Zhou, C., Tao, H., Chen, Y., Stojanovic, V., Paszke, W.: Robust point-to-point iterative learning control for constrained systems: a minimum energy approach. Int. J. Robust Nonlinear Control **32**(18), 10,139–10,161 (2022). https://doi.org/10.1002/rnc.6354
30. Stojanovic, V.: Fault-tolerant control of a hydraulic servo actuator via adaptive dynamic programming. Math Model. Control **3**(3), 181–191 (2023). https://doi.org/10.3934/mmc.2023016
31. Hayati, S.: In: Proceedings. IEEE International Conference on Robotics and Automation **3**(1986), 82–89 (1986). https://doi.org/10.1109/ROBOT.1986.1087650
32. Zheng, Y.F., Luh, J.Y.S.: In: Proceedings. 1988 IEEE International Conference on Robotics and Automation vol. 1, pp. 344–349 (1988). https://doi.org/10.1109/ROBOT.1988.12072
33. X. Song, P. Sun, S. Song, V. Stojanovic, Quantized neural adaptive finite-time preassigned performance control for interconnected nonlinear systems. Neural Comput. Appl. **35**(21), 15,429–15,446 (2023). https://doi.org/10.1007/s00521-023-08361-y
34. Alberts, T.E., Soloway, D.I., In: Proceedings. 1988 IEEE International Conference on Robotics and Automation vol.3, pp. 1490–1496 (1988). https://doi.org/10.1109/ROBOT.1988.12278
35. Walker, I.D., Freeman, R.A., Marcus, S.I.: Analysis of motion and internal loading of objects grasped by multiple cooperating manipulators. Int. J. Robot. Res. **10**(4), 396–409 (1991). https://doi.org/10.1177/027836499101000408
36. Springer Handbook of Robotics, Springer, pp. 704–705 (2008)
37. Mejia, L., Simas, H., Martins, D.: Force capability in general 3dof planar mechanisms. Mech. Mach. Theory **91**, 120–134 (2015). https://doi.org/10.1016/j.mechmachtheory.2015.04.013
38. Berghuis, H., Nijmeijer, H.: A passivity approach to controller-observer design for robots. IEEE Trans. Robot. Autom. **9**(6), 740–754 (1993). https://doi.org/10.1109/70.265918

39. Corke, P.: Robotics, Vision and Control, vol. 118, 2nd (edn.), Springer International Publishing, (2017). https://doi.org/10.1007/978-3-319-54413-7
40. ROBOTIS. Open manipulator-x (2017). https://emanual.robotis.com/docs/en/platform/openmanipulator_x/overview/

Keshab Patra is currently pursuing the Ph.D. degree in the Department of Mechanical Engineering, Indian Institute of Technology Bombay. Prior to that he was a Design Engineer at Hero MotoCorp Ltd., Jaipur, India for 1.5 year. He received the B.Tech degree from the Mechanical Engineering Department at Heritage Institute of Technology, Kolkata in 2016 and the M.Tech degree from the Department of Mechanical Engineering, Indian Institute of Technology Bombay, India in 2018. His current research includes robotics motion planning, control and multi manipulators planning.

Arpita Sinha has been a faculty in Systems and Control Engineering at Indian Institute of Technology Bombay, India since 2009. Before that, she was a postdoctoral fellow at Cranfield University, UK, for a year. She obtained her doctoral degree from the Aerospace Department at Indian Institute of Science, Bangalore, India in 2007. Her research interest lies at the intersection of robotics and control systems. The broad areas of her work include guidance and control of autonomous vehicles, multiple vehicle coordination, and distributed decision-making. She is currently working on trajectory planning for autonomous on-road driving, multi-copter control for precision agriculture, and coordination of multiple mobile manipulators.

Anirban Guha is currently a TIH-IoT Chanakya Chair Professor with the Department of Mechanical Engineering, Indian Institute of Technology Bombay, India. He received the B.Sc. (Tech) degree from Calcutta University in 1994 and M.Tech. and Ph.D. degrees from Indian Institute of technology Delhi, India in 1996 and 2002. His research interests lie in modelling of human locomotion and assistive device for human motion, remotely operated vehicles, collaborative robots, design of machine for textile industry, Artificial Intelligence and Machine Learning for structural health monitoring and image processing and impact resistance structures.