2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Distributed Co-Design of Motors and Motions for Robotic Manipulators

Zehui Lu, Yebin Wang, Yusuke Sakamoto, and Shaoshuai Mou

Abstract— **This paper studies a manipulator co-design problem of motors and motions for multiple tasks. To reduce computational burden and improve scalability as the number of tasks grows, this paper introduces a distributed co-design framework to handle the co-design process for all tasks in a distributed fashion. Moreover, this paper presents a distributed constrained optimization algorithm, which secures a unified set of design parameters for all tasks ultimately such that the total average of motor operation efficiency is optimized and the design constraints are satisfied across all tasks and motors. The distributed manner reduces the computational load by allowing each agent to solve co-design optimization solely for its designated task. A numerical simulation further verifies the proposed algorithm.**

I. Introduction

Off-the-shelf robotic manipulators typically have specified features to meet diverse user needs. These robots are typically designed to cater to a wide range of tasks, which implies optimality or sub-optimality for a particular performance index. Such an application-oriented and optimization-based robot design can lead to efficient solutions in terms of energy usage, cost-effectiveness, and productivity. The robot design is naturally multidisciplinary, involving structural and geometric design [1]–[3], battery sizing [4], kinematics [5], dynamics [2], [6], and control [7], [8]. The design objectives are also multidisciplinary, including weight [1], [9], energy consumption [10], task completion time [10], [11], etc.

The process of co-design, which addresses the interdependence and conflicts between subsystems during the design phase [12], has the potential to mitigate sub-optimal outcomes that may arise from a design process focused on individual disciplines. The concept of co-design is prevalent in numerous applications, such as general robotic module selection [12], robotic manipulators [1], [7], [13], legged robots [14]–[16], medical robots [3], soft robots [17], etc. In particular, [1]–[3] optimizes the structure, inertia, and shape of a robot. [7] co-designs the drivetrain and joint trajectories of a manipulator, where the drivetrain is parameterized by motor shaft length and gearbox ratio. [8] performs co-design of mechanical plant parameters and optimal feedback control strategies. [9] optimizes the choice of gearbox, motor drive train, and parameters for the manipulator's links. [10], [18] perform simultaneous optimization for both trajectory and controller design. [19] co-optimizes for motions, robot physical parameters, and motor selection. [20] presents a bi-level optimization framework to determine the optimal hardware parameters for a legged robot and an energy-efficient trajectory for a given task, operating at two distinct levels. Moreover, there has been a growing interest in task-specific robot co-design derived from high-level user specifications. Existing work includes co-design of motion and physical parameters for a single task [1], [2], [7]–[9], [19] and for multiple tasks [10], [18].

A primary obstacle in robot co-design is the significant computational load [14]. In practice, robot co-design needs to consider the specifications of multiple customers at once, due to the nature of manufacturing. Thus, co-design becomes more complicated when multiple objectives must be considered. To tackle this issue, [10], [18] propose a centralized bi-level stochastic programming framework for the co-design under multiple tasks. This framework considers task versatility by incorporating an overall expected cost function that accounts for the probability distribution of task occurrence rates. Specifically, at the outer level, the robot design is optimized by minimizing the average cost across all tasks. At the inner level, the robot's motion is optimized independently for each task. Such a centralized framework aggregates information from all tasks and can pose computational challenges as the number of tasks grows, no matter whether the design parameters and the motion are determined simultaneously [3] or sequentially [10], [18]. Different from the centralized bi-level framework above, [21] proposes a consensus-based distributed bi-level optimization framework, which cooperatively adjusts the motion from each agent's trajectory planner to further optimize an additional performance index.

Motivated by the distributed bi-level framework mentioned earlier, this paper aims to co-design the motions (joint position and velocity trajectories) and motor design parameters for a robotic manipulator with n degree-of-freedom (DOF) to handle multiple tasks in a distributed fashion. To achieve this, each task is conceptualized as an agent within a network. Given a task and arbitrary motor design parameters, a trajectory of joints' angular positions and velocities can be computed by a local trajectory planner that aligns with the task requirements. Subsequently, a local loss function is used to measure the efficiency of each agent's motors based on their respective trajectory. Due to manufacturing limitations, a unified set of design parameters is ultimately required to

Z. Lu and S. Mou are with the School of Aeronautics and Astronautics, Purdue University, IN 47907, USA. Email: {lu846,mous}@purdue.edu This work was done while Z. Lu was an intern at Mitsubishi Electric Research Laboratories (MERL).

Y. Wang is with MERL, Cambridge, MA 02139, USA. Email: yebinwang@ieee.org

Y. Sakamoto was with MERL and is with Advanced Technology R&D Center, Mitsubishi Electric Corporation, Amagasaki City, Japan. Email: sakamoto.yusuke@df.mitsubishielectric.co.jp

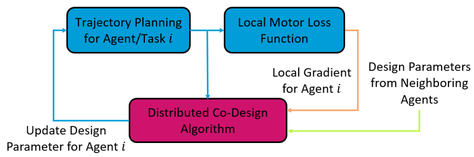

Fig. 1. The distributed co-design framework.

fulfill all tasks, thereby optimizing the global average of each task's local loss. This requirement of common design parameters can be posed as a consensus across the network.

Therefore, as illustrated in Fig. 1, this paper introduces a distributed co-design framework and proposes a distributed optimization algorithm based on the combination of the alternating direction method of multipliers (ADMM) and the augmented Lagrangian method (ALM). The aim is to compute optimal and unified design parameters, optimizing the global average of local loss functions in a distributed and iterative manner to effectively fulfill multiple tasks. The distributed manner reduces the computational load by allowing each agent to solve co-design optimization solely for its designated task. This framework ensures the scalability of the multiple-task co-design as the number of tasks grows. The contributions of this paper are summarized as follows:

1) A distributed framework for manipulator co-design of motors and motions with multiple tasks;
2) A distributed constrained optimization algorithm to solve the distributed co-design problem.

Notations. The non-negative integer set is denoted by $\mathbb{Z}_+$. For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, $\boldsymbol{x} \leq \boldsymbol{y}$ indicates element-wise inequality. Let $\mathrm{col}\{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_a\}$ denote a column stack of elements $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_a$, which may be scalars, vectors or matrices, i.e. $\mathrm{col}\{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_a\} \triangleq \begin{bmatrix} \boldsymbol{v}_1{}^\top & \cdots & \boldsymbol{v}_a{}^\top \end{bmatrix}^\top$. Let $\otimes$ denote the Kronecker product. Let $\mathbf{0}_n, \mathbf{1}_n \in \mathbb{R}^n$ denote a zero and an one vector. Let $I_n \in \mathbb{R}^{n \times n}$ denote an identity matrix.

II. System Modeling

This section introduces the modeling of an arbitrary surface permanent magnet synchronous motor's (SPMSM) dynamics and an arbitrary n-DOF open-chain manipulator dynamics with n SPMSMs.

A. SPMSM Modeling and Design Parametrization

This subsection presents the dynamical modeling of SPMSM with several motor design variables. The motor design variables are summarized in Table I. Fig. 2 further illustrates the physical meaning of the design variables, where the axial length l is not shown. Denote an arbitrary motor's design variables as

$$\boldsymbol{\beta} \triangleq \mathrm{col}\{l, r_{\mathrm{ro}}, r_{\mathrm{so}}, h_{\mathrm{m}}, h_{\mathrm{sy}}, w_{\mathrm{tooth}}, b_0\} \in \mathbb{R}^7.$$

TABLE I
SPMSM Design Parameters

Parameter	Description	Parameter	Description
l	Axial length of core	h_{sy}	Stator yoke
r_{ro}	Outer radius of rotor	w_{tooth}	Width of tooth
r_{so}	Outer radius of stator	b_0	Slot opening
h_{m}	Height of magnet		

Units of all design parameters are mm

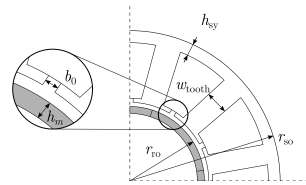

Fig. 2. The cross-section of an SPMSM design with parameterization.

Given the motor design parameters for one motor, the magnetic equivalent circuit (MEC) modeling technique is used to compute the necessary parameters for the motor dynamics analytically. The derivation details are summarized in Appendix A. The dynamic model of an arbitrary SPMSM can be written as follows [22]:

$$\frac{di_{\mathrm{d}}}{dt} = -\tfrac{R}{L_{\mathrm{d}}} i_{\mathrm{d}} + p\omega i_{\mathrm{q}} + \tfrac{u_{\mathrm{d}}}{L_{\mathrm{d}}}, \tag{1a}$$

$$\frac{di_{\mathrm{q}}}{dt} = -\tfrac{R}{L_{\mathrm{q}}} i_{\mathrm{q}} - p\omega(i_{\mathrm{d}} + \tfrac{\Phi_{\mathrm{m}}}{L_{\mathrm{q}}}) + \tfrac{u_{\mathrm{q}}}{L_{\mathrm{q}}}, \tag{1b}$$

where i_{d} and i_{q} are the currents in the d- and q-axis, respectively; u_{d} and u_{q} are the voltages in the d- and q-axis, respectively; ω is the motor's angular velocity. p is referred to Appendix A. $R, L_{\mathrm{d}}, L_{\mathrm{q}}, \Phi_{\mathrm{m}}$ are calculated by (22), (23) and (26) in Appendix A with $\boldsymbol{\beta}$. The motor design variables are subject to the following constraints:

$$l \in [20, 100], \; r_{\mathrm{ro}} \in [10, 100], \tag{2a}$$

$$r_{\mathrm{so}} \in [10, 100], \; h_{\mathrm{m}} \in [1, 5], \; h_{\mathrm{sy}} \in [5, 10], \tag{2b}$$

$$w_{\mathrm{tooth}} \in [5, 20], \; b_0 \in [1, 10], \tag{2c}$$

$$h_{\mathrm{ss}} > 0 \text{ mm}, \quad D_{\mathrm{wire}} \geq 0.6 \text{ mm}, \quad k_{\mathrm{C}} > 0, \tag{2d}$$

$$0 < \arcsin(\tfrac{w_{\mathrm{tooth}}}{2(r_{\mathrm{ro}}+\delta)}) + \arcsin(\tfrac{b_0}{2(r_{\mathrm{ro}}+\delta)}) \leq \tfrac{\pi}{Q}, \tag{2e}$$

$$0 \text{ kg} < m_{\mathrm{stator}} + m_{\mathrm{rotor}} \leq 3 \text{ kg}, \; m_{\mathrm{stator}} > 0 \text{ kg}, \tag{2f}$$

$$0 \text{ T} < \tfrac{k_{\mathrm{p}}\Phi_1}{w_{\mathrm{tooth}} l} \leq 1.5 \text{ T}, \quad 0 \text{ T} < \tfrac{k_{\mathrm{p}}\Phi_1}{\sqrt{3} h_{\mathrm{sy}} l} \leq 1.5 \text{ T}, \tag{2g}$$

where (2d) describe the minimal slot height, motor's minimal wire diameter, and minimal Carter's coefficient; (2e) and (2f) describe the tooth width bound and the motor weight bound, respectively; (2g) describes the magnetic flux bounds in the tooth and the stator yoke. δ and $h_{\mathrm{ss},j}$, $D_{\mathrm{wire},j}$, $m_{\mathrm{stator},j}$, $m_{\mathrm{rotor},j}$, Φ_1, k_{p}, k_{C} can be obtained and calculated by (16) - (21) and (24) - (27) of Appendix A. The motor is additionally subject to some operational constraints for all t:

$$-3 \text{ A} \leq i_{\mathrm{d}} \leq 0 \text{ A}, \qquad -3 \text{ A} \leq i_{\mathrm{q}} \leq 3 \text{ A}, \tag{3a}$$

$$-100 \text{ V} \leq u_{\mathrm{d}} \leq 100 \text{ V}, -100 \text{ V} \leq u_{\mathrm{q}} \leq 100 \text{ V}. \tag{3b}$$

B. n-DOF Open-Chain Manipulator Dynamics

Assume that the flexibility and the backlashes of the gears can be ignored, and the $(j+1)$-th motor of a manipulator is rigidly attached to the j-th link, i.e., its stator is rigidly attached to the j-th link and its rotor is coupled with the $j+1$-th link. Following the standard modeling approach in [23, Chapter 8], a general dynamic model for an arbitrary n-DOF open-chain manipulator can be written as

$$M(\boldsymbol{\theta},\boldsymbol{\beta})\ddot{\boldsymbol{\theta}} + C(\boldsymbol{\theta},\dot{\boldsymbol{\theta}},\boldsymbol{\beta})\dot{\boldsymbol{\theta}} + G(\boldsymbol{\theta},\boldsymbol{\beta}) = \boldsymbol{\tau}, \tag{4}$$

where $\boldsymbol{\theta},\dot{\boldsymbol{\theta}},\ddot{\boldsymbol{\theta}} \in \mathbb{R}^n$ are the angular positions [rad], velocities [rad/s^2], and accelerations [rad/s^2] of n joints, respectively; $M(\boldsymbol{\theta},\boldsymbol{\beta}), C(\boldsymbol{\theta},\dot{\boldsymbol{\theta}},\boldsymbol{\beta}) \in \mathbb{R}^{n\times n}$ and $G(\boldsymbol{\theta},\boldsymbol{\beta}) \in \mathbb{R}^n$ are the inertia-related matrix, Coriolis matrix, and gravitational force with the parameterization of SPMSM under $\boldsymbol{\beta}$, respectively; $\boldsymbol{\tau} \in \mathbb{R}^n$ are torques applied on n joints, where its j-th element is the torque applied on the j-th joint and is determined by (32), Appendix B.

The articulated-body algorithm (ABA) [24, Chapter 7.3] is used to compute the manipulator's forward dynamics with zero tip force and a constant gravitational acceleration $g = 9.81$ m/s^2. Details are summarized in [25, Algorithm 1]. The constraints on $\boldsymbol{\theta}$ and $\dot{\boldsymbol{\theta}}$ are given by:

$$\underline{\boldsymbol{\theta}} \leq \boldsymbol{\theta} \leq \overline{\boldsymbol{\theta}}, \quad \underline{\dot{\boldsymbol{\theta}}} \leq \dot{\boldsymbol{\theta}} \leq \overline{\dot{\boldsymbol{\theta}}}, \tag{5}$$

where $\underline{\boldsymbol{\theta}}, \overline{\boldsymbol{\theta}}, \underline{\dot{\boldsymbol{\theta}}}, \overline{\dot{\boldsymbol{\theta}}} \in \mathbb{R}^n$ denotes the position lower and upper bound, velocity lower and upper bound, respectively.

Remark 1. *Each task may require a different payload on the end-of-effector, which can be treated as a rigid body attached to the final link n. Given the mass, center of mass and rotational inertia of a payload, one can compute its spatial rigid-body inertia matrix and then add it to the spatial inertia matrix of link n. The manipulator's forward dynamics can still be computed by ABA [24, Chapter 7.3].*

C. Complete Manipulator Dynamics

Denote $i_{\mathrm{d},j}$ and $u_{\mathrm{d},j}$ as the current and voltage in the d-axis for the j-th motor of an arbitrary manipulator. Denote the state of all the motors by $\boldsymbol{x}_{\mathrm{m}} \triangleq \mathrm{col}\{i_{\mathrm{d},1},\cdots,i_{\mathrm{d},n},i_{\mathrm{q},1},\cdots,i_{\mathrm{q},n},\} \in \mathbb{R}^{2n}$; similarly denote the control of all motors by $\boldsymbol{u}_{\mathrm{m}} \triangleq \mathrm{col}\{u_{\mathrm{d},1},\cdots,u_{\mathrm{d},n},u_{\mathrm{q},1},\cdots,u_{\mathrm{q},n},\} \in \mathbb{R}^{2n}$. Combining the n-DOF manipulator dynamics (4) with n motor's dynamics (1) and the connection among motor states, controls, velocities, and torques (28)-(32) of Appendix B, the complete dynamics for an arbitrary manipulator are written as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{\beta}), \tag{6}$$

where $\boldsymbol{x} \triangleq \mathrm{col}\{\boldsymbol{\theta},\dot{\boldsymbol{\theta}},\boldsymbol{x}_{\mathrm{m}}\} \in \mathbb{R}^{4n}$ and $\boldsymbol{u} \triangleq \boldsymbol{u}_{\mathrm{m}} \in \mathbb{R}^{2n}$.

D. SPMSM Design Optimization Formulation

This subsection formulates the SPMSM design as an optimization problem, which aims to optimize operation efficiency given a specific task and the respective motor trajectory. Given the task and the complete dynamics (6), a trajectory planner, which will be defined in (9), can generate a trajectory of optimal open-loop states $\boldsymbol{x}^*(t)$ and controls $\boldsymbol{u}^*(t)$. Then the desired trajectory of joint angular positions $\boldsymbol{\theta}_{\mathrm{des}}(t)$ and velocities $\dot{\boldsymbol{\theta}}_{\mathrm{des}}(t)$ for all motors can be directly obtained from $\boldsymbol{x}^*(t)$ and $\boldsymbol{u}^*(t)$. From the description below (32) of Appendix B, a trajectory of each motor's velocity $\omega(t)$ and motor torque $\tau_{\mathrm{m}}(t)$ can be directly obtained. Consequently, each motor's maximum (magnitude) velocity $\omega_{\max}$ and torque $\tau_{\max}$ are determined.

Then for each motor, given the operational data $\boldsymbol{\xi} \triangleq \{\boldsymbol{x}^*(t),\boldsymbol{u}^*(t),\ \forall t\}$, one can measure the probability of each grid $(\tau_{\mathrm{m}},\omega)$ within the map range $A \triangleq [0,\tau_{\max}]\times[0,\omega_{\max}]$ by a two-dimensional probability density function (PDF) $q(\tau_{\mathrm{m}},\omega;\boldsymbol{\beta})$. The details to generate a PDF are summarized in Algorithm 1, where $N_\tau, N_\omega \in \mathbb{Z}_+$ are the grid numbers for the discretization of motor torque and velocity, respectively. Line 9 of Algorithm 1 converts a single operational data point $\{\boldsymbol{x}^*(t),\boldsymbol{u}^*(t)\}$ at a time instance t to a grid $(\tau_{\mathrm{m}},\omega)$, i.e. (28)-(32) of Appendix B.

Next, the efficiency $\eta(\tau_{\mathrm{m}},\omega;\boldsymbol{\beta})$ at each grid $(\tau_{\mathrm{m}},\omega)$ can be calculated by (33), Appendix B. Thus the design optimization for one motor is written as:

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & \iint_A q(\tau_{\mathrm{m}},\omega;\boldsymbol{\beta})(1-\eta(\tau_{\mathrm{m}},\omega;\boldsymbol{\beta}))d\omega d\tau_{\mathrm{m}} \\ \text{s.t.} \quad & \boldsymbol{h}_{\mathrm{d}}(\boldsymbol{\beta}) \leq \boldsymbol{0}, \\ & \boldsymbol{h}_{\mathrm{o}}(\boldsymbol{\beta},\tau_{\mathrm{m}},\omega) \leq \boldsymbol{0},\ \forall(\tau_{\mathrm{m}},\omega)\in A, \end{aligned} \tag{7}$$

where $\iint_A$ denotes the double integration $\int_0^{\tau_{\max}}\int_0^{\omega_{\max}}$ among $A \triangleq [0,\tau_{\max}]\times[0,\omega_{\max}]$; $\tau_{\max}$ and $\omega_{\max}$ are determined by $\boldsymbol{\xi}$; the motor design constraints $\boldsymbol{h}_{\mathrm{d}}(\boldsymbol{\beta})$ are summarized as (2); the motor operational constraints $\boldsymbol{h}_{\mathrm{o}}(\boldsymbol{\xi},\boldsymbol{\beta})$ are summarized as (3), where $i_{\mathrm{d}}, i_{\mathrm{q}}, u_{\mathrm{d}}, u_{\mathrm{q}}$ are determined by (35) given each $(\tau_{\mathrm{m}},\omega)$ within A. Note that evaluating the efficiency of a motor operation PDF is more robust and numerically stable than of a specific trajectory.

III. Distributed Co-Design Formulation

This section introduces the formulation of a distributed consensus-based co-design problem. Consider a network of N_a agents labeled as $\mathcal{V} = \{1,\cdots,N_a\}$, where each agent i determines the design of manipulator i based on a given task i. Agent i can receive information from its neighbor set $\mathcal{N}_i$. $\mathbb{G} = \{\mathcal{V},\mathcal{E}\}$ denotes an undirected graph such that an undirected edge $(i,r) \in \mathcal{E}$ if and only if i and r are neighbors. $\mathcal{L} \in \mathbb{R}^{N_a\times N_a}$ denotes the Laplacian matrix of $\mathbb{G}$. $\bar{\mathcal{L}} \triangleq \mathcal{L}\otimes I_{7n}$. Denote the total number of edges within $\mathbb{G}$ as m. Define the oriented incidence matrix of $\mathbb{G}$ denoted by $H \in \mathbb{R}^{m\times N_a}$ such that its entry at the k-th row and the r-th column is 1 if edge k is an incoming edge to node r; -1 if edge k is an outgoing edge to node r; and 0 elsewhere. Note that for undirected graphs, the direction for each edge could be arbitrary. Denote $\bar{H} \triangleq H\otimes I_{7n}$.

Algorithm 1: 2D Speed-Torque PDF Generation

Input: N_τ, N_ω, $\boldsymbol{\xi}$, $N_\mathrm{s}=0$, ω_max, τ_max

$q \leftarrow \mathrm{zeros}(N_\tau, N_\omega)$ // Initialize the PDF
$\boldsymbol{v}_\omega \leftarrow \mathrm{linspace}(0, \omega_\mathrm{max}, N_\omega + 1)$
$\boldsymbol{v}_\tau \leftarrow \mathrm{linspace}(0, \tau_\mathrm{max}, N_\tau + 1)$
for $i_\omega = 1$ *to* N_ω **do**
  $\omega_1 \leftarrow \boldsymbol{v}_\omega[i_\omega]$, $\omega_2 \leftarrow \boldsymbol{v}_\omega[i_\omega+1]$
  for $i_\tau = 1$ *to* N_τ **do**
    $\tau_1 \leftarrow \boldsymbol{v}_\tau[i_\tau]$, $\tau_2 \leftarrow \boldsymbol{v}_\tau[i_\tau+1]$, $c \leftarrow 0$
    for *each* datapoint *in* $\boldsymbol{\xi}$ **do**
      $\omega, \tau_\mathrm{m} \leftarrow \mathrm{parse}(\mathrm{datapoint})$
      if $\tau_\mathrm{m} \in [\tau_1, \tau_2)$ and $\omega_\mathrm{m} \in [\omega_1, \omega_2)$ **then**
        $c \leftarrow c+1$, $N_\mathrm{s} \leftarrow N_\mathrm{s}+1$
      if $i_\tau == N_\tau$ and $\tau_\mathrm{m} \geq \tau_2$ and $\omega_\mathrm{m} \in [\omega_1, \omega_2)$ **then** $c \leftarrow c+1$, $N_\mathrm{s} \leftarrow N_\mathrm{s}+1$
    $q[i_\tau, i_\omega] \leftarrow c$

Output: $q \leftarrow q/N_\mathrm{s}$ // Normalize the PDF

The SPMSM design variables for manipulator-i's j-th motor are denoted by $l_{i,j}$, $r_{\mathrm{ro},i,j}$, $r_{\mathrm{so},i,j}$, $h_{\mathrm{m},i,j}$, $h_{\mathrm{sy},i,j}$, $w_{\mathrm{tooth},i,j}$, $b_{0,i,j}$. And further denote

$$\boldsymbol{\beta}_{i,j} \triangleq \mathrm{col}\{l_{i,j}, r_{\mathrm{ro},i,j}, r_{\mathrm{so},i,j}, h_{\mathrm{m},i,j}, h_{\mathrm{sy},i,j}, w_{\mathrm{tooth},i,j}, b_{0,i,j}\},$$
$$\boldsymbol{\beta}_i \triangleq \mathrm{col}\{\boldsymbol{\beta}_{i,1}, \cdots, \boldsymbol{\beta}_{i,n}\} \in \mathbb{R}^{7n},$$

where $\boldsymbol{\beta}_i$ indicates all the motor design parameters of manipulator i. Similar as (6), the complete dynamics for manipulator i are written as

$$\dot{\boldsymbol{x}}_i = \boldsymbol{f}_i(\boldsymbol{x}_i(t), \boldsymbol{u}_i(t), \boldsymbol{\beta}_i), \tag{8}$$

where $\boldsymbol{x}_i \triangleq \mathrm{col}\{\boldsymbol{\theta}_i, \dot{\boldsymbol{\theta}}_i, \boldsymbol{x}_{\mathrm{m},i}\} \in \mathbb{R}^{4n}$ and $\boldsymbol{u}_i \triangleq \boldsymbol{u}_{\mathrm{m},i} \in \mathbb{R}^{2n}$.

Given a task, a trajectory planning optimization can return the optimal trajectories of states and controls that align with the task requirements:

$$\begin{aligned}
\min_{\boldsymbol{x}_i(t), \boldsymbol{u}_i(t), t_{\mathrm{f},i}} \quad & J_i(\boldsymbol{x}_i(t), \boldsymbol{u}_i(t), t_{\mathrm{f},i}) && \text{(9a)}\\
\text{s.t.} \quad & \dot{\boldsymbol{x}}_i = \boldsymbol{f}_i(\boldsymbol{x}_i(t), \boldsymbol{u}_i(t), \boldsymbol{\beta}_i), && \text{(9b)}\\
& \forall t \in [0, t_{\mathrm{f},i}] \text{ with given } \boldsymbol{x}_i(0), && \text{(9c)}\\
& \text{constraints (3), (5)}, \ \forall t, && \text{(9d)}\\
& \boldsymbol{\theta}_i(t_{\mathrm{f},i}) = \boldsymbol{\theta}_{\mathrm{des},i}, \ \dot{\boldsymbol{\theta}}_i(t_{\mathrm{f},i}) = \boldsymbol{0}, && \text{(9e)}
\end{aligned}$$

where $t_{\mathrm{f},i} > 0$ denotes the final time for trajectory planning, which could either be a decision variable or a fixed prescribed parameter; $\boldsymbol{\theta}_{\mathrm{des},i}$ denotes a prescribed desired final position. Note that (9) adopts the most general form, which represents time-optimal trajectory planning, trajectory tracking, or energy-optimal trajectory planning. For the latter two cases, $t_{\mathrm{f},i}$ is a fixed prescribed parameter. Given a particular value of $\boldsymbol{\beta}_i$, the optimal states $\boldsymbol{x}_i^*(t)$, controls $\boldsymbol{u}_i^*(t)$, and final time $t_{\mathrm{f},i}^*$ (if applicable) optimize the cost function J_i. For notational simplicity, let $\boldsymbol{\xi}_i \triangleq \{\boldsymbol{x}_i^*(t), \boldsymbol{u}_i^*(t), \forall t, \ t_{\mathrm{f},i}^*\}$ denote the optimal trajectory of manipulator i given the task.

Then similar to (7), the distributed manipulator co-design problem can be rewritten as:

$$\begin{aligned}
\min_{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{N_a}} \quad & \textstyle\sum_{i=1}^{N_a} \ell_i(\boldsymbol{\xi}_i, \boldsymbol{\beta}_i) && \text{(10a)}\\
\text{s.t.} \quad & \boldsymbol{\beta}_1 = \cdots = \boldsymbol{\beta}_{N_a}, && \text{(10b)}\\
& \boldsymbol{\xi}_i \text{ obtained from (9) given } \boldsymbol{\beta}_i, && \text{(10c)}\\
& \boldsymbol{h}_{\mathrm{d},i}(\boldsymbol{\beta}_i) \leq \boldsymbol{0}, && \text{(10d)}\\
& \boldsymbol{h}_{\mathrm{o},i}(\boldsymbol{\beta}_i, \boldsymbol{\tau}_{\mathrm{m},i}, \boldsymbol{\omega}_i) \leq \boldsymbol{0}, && \text{(10e)}\\
& \forall \ (\boldsymbol{\tau}_{\mathrm{m},i}, \boldsymbol{\omega}_i) \in A_i, \ \forall i \in \mathcal{V}, && \text{(10f)}
\end{aligned}$$

where $\ell_i(\cdot)$ is a local loss function that measures the total operational efficiency loss of manipulator-i's n motors, i.e.

$$\begin{aligned}
\ell_i(\boldsymbol{\xi}_i, \boldsymbol{\beta}_i) \triangleq \textstyle\sum_{j=1}^{n} \iint_{A_{ij}} & q_{i,j}(\tau_{\mathrm{m},i,j}, \omega_{i,j}; \boldsymbol{\beta}_{i,j}) \cdot \\
& (1 - \eta(\tau_{\mathrm{m},i,j}, \omega_{i,j}; \boldsymbol{\beta}_{i,j})) d\omega_{i,j} d\tau_{\mathrm{m},i,j}.
\end{aligned} \tag{11}$$

$\tau_{\mathrm{max},i,j}, \omega_{\mathrm{max},i,j}$ are determined by $\boldsymbol{\xi}_i$. $\boldsymbol{h}_{\mathrm{d},i}(\cdot)$ denotes a column stack of $\boldsymbol{h}_\mathrm{d}(\boldsymbol{\beta}_{i,j})$ defined in (7), $\forall j = 1, \cdots, n$. $\boldsymbol{h}_{\mathrm{o},i}(\cdot)$ denotes a column stack of $\boldsymbol{h}_\mathrm{o}$ defined in (7) with $\boldsymbol{\tau}_{\mathrm{m},i} \triangleq \mathrm{col}\{\tau_{\mathrm{m},i,1}, \cdots, \tau_{\mathrm{m},i,n}\}$, $\boldsymbol{\omega}_i \triangleq \mathrm{col}\{\omega_{i,1}, \cdots, \omega_{i,n}\}$, $A_i \triangleq A_{i,1} \times \cdots A_{i,n}$, and $A_{i,j} \triangleq [0, \tau_{\mathrm{max},i,j}] \times [0, \omega_{\mathrm{max},i,j}]$. The consensus constraint (10b) ensures that there will be only one set of design parameters in the solution.

IV. Distributed Co-Design Algorithm

Combining ADMM and ALM, this section proposes a distributed constrained optimization algorithm to solve (10) iteratively. Originally, the augmented Lagrangian of (10) is not separable for each $\boldsymbol{\beta}_i$ and hence the gradient descent of this Lagrangian w.r.t. $\boldsymbol{\beta}$ cannot be distributed to each $\boldsymbol{\beta}_i$. Thus, an equivalent augmented Lagrangian is presented such that the update rule is distributed for $\boldsymbol{\beta}_i$. First, an assumption and a theorem are introduced.

Assumption 1. *The undirected graph $\mathbb{G}$ is connected.*

Theorem 1. *Let Assumption 1 hold. Denote $\hat{\boldsymbol{\beta}} \triangleq$ col$\{\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_{N_a}\} \in \mathbb{R}^{7nN_a}$. The consensus constraint (10b) is equivalent to $\bar{\mathcal{L}}\hat{\boldsymbol{\beta}} = \boldsymbol{0}$ and further $\bar{H}\hat{\boldsymbol{\beta}} = \boldsymbol{0}$.*

Proof. By Assumption 1, the graph $\mathbb{G}$ is connected. Then the consensus constraint $\boldsymbol{\beta}_1 = \cdots = \boldsymbol{\beta}_{N_a}$ holds if and only if $\bar{\mathcal{L}}\hat{\boldsymbol{\beta}} = \boldsymbol{0}$. Given $\mathcal{L} = H^\top H$, $\bar{\mathcal{L}} \equiv H^\top H \otimes I_{7n} \equiv (H^\top \otimes I_{7n})(H \otimes I_{7n}) \equiv \bar{H}^\top \bar{H}$. Then the consensus constraint holds if and only if $\bar{H}^\top \bar{H}\hat{\boldsymbol{\beta}} = \boldsymbol{0}$. Apparently $\bar{H}\hat{\boldsymbol{\beta}} = \boldsymbol{0} \Rightarrow \bar{H}^\top \bar{H}\hat{\boldsymbol{\beta}} = \boldsymbol{0}$. Since $\mathbb{G}$ is connected, from [26, Theorem 8.3.1], there is one connected component of $\mathbb{G}$, and hence ker $H = \boldsymbol{1}$. Thus, ker $H^\top = \boldsymbol{0}$ and ker $\bar{H}^\top = \boldsymbol{0}$. By the kernel's definition, $\bar{H}^\top \bar{H}\hat{\boldsymbol{\beta}} = \boldsymbol{0} \Rightarrow \bar{H}\hat{\boldsymbol{\beta}} = \boldsymbol{0}$. Thus, $\bar{H}^\top \bar{H}\hat{\boldsymbol{\beta}} = \boldsymbol{0} \Leftrightarrow \bar{H}\hat{\boldsymbol{\beta}} = \boldsymbol{0}$. Therefore, the consensus constraint holds $\Leftrightarrow \bar{\mathcal{L}}\hat{\boldsymbol{\beta}} = \boldsymbol{0} \Leftrightarrow \bar{H}\hat{\boldsymbol{\beta}} = \boldsymbol{0}$. ■

Define $\boldsymbol{g}_i(\boldsymbol{\beta}_i, \boldsymbol{\tau}_{\mathrm{m},i}, \boldsymbol{\omega}_i) \triangleq \max(\boldsymbol{h}_{\mathrm{o},i}(\boldsymbol{\beta}_i, \boldsymbol{\tau}_{\mathrm{m},i}, \boldsymbol{\omega}_i), 0)^2$, where $\max(\cdot)$ and square are applied element-wise. Then the inequality constraint (10e) can be converted to an equality constraint, i.e. $\boldsymbol{g}_i(\boldsymbol{\beta}_i, \boldsymbol{\tau}_{\mathrm{m},i}, \boldsymbol{\omega}_i) = \boldsymbol{0}$. Together with

Theorem 1, (10) is equivalent to an equality-constrained optimization:

$$\min_{\hat{\boldsymbol{\beta}} \in \mathcal{B} \subset \mathbb{R}^{7nN_a}} \quad \textstyle\sum_{i=1}^{N_a} \ell_i(\boldsymbol{\xi}_i, \boldsymbol{\beta}_i) \tag{12a}$$

$$\text{s.t.} \quad \bar{\mathcal{L}}\hat{\boldsymbol{\beta}} = \mathbf{0}, \tag{12b}$$

$$\boldsymbol{g}_i(\boldsymbol{\beta}_i, \boldsymbol{\tau}_{\mathrm{m},i}, \boldsymbol{\omega}_i) = \mathbf{0}, \tag{12c}$$

where $\mathcal{B} \triangleq \mathcal{B}_1 \times \cdots \times \mathcal{B}_{N_a}$ represents the feasible set among all agents, and $\mathcal{B}_i \subset \mathbb{R}^{7n}$ represents agent-i's feasible set constrained by $\boldsymbol{h}_{\mathrm{d},i}(\boldsymbol{\beta}_i) \leq \mathbf{0}$ in (10d). Introduce the Lagrangian multipliers $\boldsymbol{\lambda}_{\mathrm{L}}$ and $\boldsymbol{\lambda}_i$ associated with (12b) and (12c), respectively. $\boldsymbol{\lambda}_{\mathrm{L}} \triangleq \mathrm{col}\{\boldsymbol{\lambda}_{\mathrm{L},1}, \cdots, \boldsymbol{\lambda}_{\mathrm{L},N_a}\} \in \mathbb{R}^{7nN_a}$, where $\boldsymbol{\lambda}_{\mathrm{L},i} \in \mathbb{R}^{7n}$; with the same dimension, denote $\boldsymbol{\lambda} \triangleq \mathrm{col}\{\boldsymbol{\lambda}_1, \cdots, \boldsymbol{\lambda}_{N_a}\}$. Similar to [27], with $\bar{\mathcal{L}}\hat{\boldsymbol{\beta}} = \mathbf{0} \Leftrightarrow \bar{H}\hat{\boldsymbol{\beta}} = \mathbf{0}$ from Theorem 1, the augmented Lagrangian associated with (12) is equivalent to

$$\begin{aligned} L(\hat{\boldsymbol{\beta}}, \boldsymbol{\lambda}_{\mathrm{L}}, \boldsymbol{\lambda}) \triangleq\ & \boldsymbol{\lambda}_{\mathrm{L}}^\top \bar{\mathcal{L}}\hat{\boldsymbol{\beta}} + \tfrac{\rho_1}{2}||\bar{H}\hat{\boldsymbol{\beta}}||^2 + \textstyle\sum_{i=1}^{N_a} \\ & (\ell_i(\boldsymbol{\xi}_i, \boldsymbol{\beta}_i) + \boldsymbol{\lambda}_i^\top \boldsymbol{g}_i(\cdot) + \tfrac{\rho_2}{2}||\boldsymbol{g}_i(\cdot)||^2), \end{aligned} \tag{13}$$

where $\rho_1, \rho_2 > 0$ are arbitrary constants. Denote $k \in \mathbb{Z}_+$ as the iteration index. To compute the optimal $\hat{\boldsymbol{\beta}}$, the gradient descent in $\hat{\boldsymbol{\beta}}$ and gradient ascent in $\boldsymbol{\lambda}_{\mathrm{L}}$ and $\boldsymbol{\lambda}$ can be applied in an ALM fashion [28]. Thus, with the fact that $\bar{\mathcal{L}}^\top = \bar{\mathcal{L}}$, the iterative update rule is given by

$$\hat{\boldsymbol{\beta}}(k+1) = \hat{\boldsymbol{\beta}}(k) - \alpha(\bar{\mathcal{L}}\boldsymbol{\lambda}_{\mathrm{L}}(k) + \rho_1\bar{\mathcal{L}}\hat{\boldsymbol{\beta}}(k) + (*)), \tag{14a}$$

$$\boldsymbol{\lambda}_{\mathrm{L}}(k+1) = \boldsymbol{\lambda}_{\mathrm{L}}(k) + \alpha\bar{\mathcal{L}}\hat{\boldsymbol{\beta}}(k+1), \tag{14b}$$

$$\boldsymbol{\lambda}_i(k+1) = \boldsymbol{\lambda}_i(k) + \alpha\boldsymbol{g}_i(\boldsymbol{\beta}_i(k+1), \cdot), \tag{14c}$$

where $(*)$ indicates the partial derivative of the summation term in (13) w.r.t. $\hat{\boldsymbol{\beta}}$; $\alpha > 0$ is a step size. Then the update (14) is naturally distributed and can be decomposed as:

$$\begin{aligned} \tilde{\boldsymbol{\beta}}_i(k+1) = \boldsymbol{\beta}_i(k) & - \alpha \textstyle\sum_{r\in\mathcal{N}_i}(\boldsymbol{\lambda}_{\mathrm{L},i}(k) - \boldsymbol{\lambda}_{\mathrm{L},r}(k) \\ & + \rho_1\boldsymbol{\beta}_i(k) - \rho_1\boldsymbol{\beta}_r(k)) - \alpha \frac{\partial \ell_i(\boldsymbol{\xi}_i, \boldsymbol{\beta}_i)}{\partial \boldsymbol{\beta}_i}^\top \\ & - \alpha \frac{\partial \boldsymbol{g}_i(\cdot)}{\partial \boldsymbol{\beta}_i}^\top \boldsymbol{\lambda}_i - \alpha\rho_2 \frac{\partial \boldsymbol{g}_i(\cdot)}{\partial \boldsymbol{\beta}_i}^\top \boldsymbol{g}_i(\cdot), \end{aligned} \tag{15a}$$

$$\boldsymbol{\beta}_i(k+1) = \arg\min_{\boldsymbol{x}\in\mathcal{B}_i} ||\tilde{\boldsymbol{\beta}}_i(k+1) - \boldsymbol{x}||_2, \tag{15b}$$

$$\begin{aligned} \boldsymbol{\lambda}_{\mathrm{L},i}(k+1) = \boldsymbol{\lambda}_{\mathrm{L},i}(k) & + \alpha \textstyle\sum_{r\in\mathcal{N}_i}(\boldsymbol{\beta}_i(k+1), \\ & - \boldsymbol{\beta}_r(k+1)), \end{aligned} \tag{15c}$$

$$\boldsymbol{\lambda}_i(k+1) = \boldsymbol{\lambda}_i(k) + \alpha\boldsymbol{g}_i(\boldsymbol{\beta}_i(k+1), \cdot), \tag{15d}$$

where $\frac{\partial \ell_i(\boldsymbol{\xi}_i, \boldsymbol{\beta}_i)}{\partial \boldsymbol{\beta}_i}$ and $\frac{\partial \boldsymbol{g}_i(\cdot)}{\partial \boldsymbol{\beta}_i}$ are given by the definition in (11), (10) and (7). Based on the update rule (15), the distributed co-design algorithm is summarized in Algorithm 2, where the content within **parfor** is executed by each agent distributedly.

Remark 2. *The motor design constraints (10d) are enforced as a projection instead of inequality constraints because the projection ensures the feasibility of trajectory planning given updated parameters $\boldsymbol{\beta}_i(k+1)$ for every iteration k.*

Algorithm 2: Distributed Co-Design Algorithm

Input: ρ_1, ρ_2, N_τ, N_ω, $N_{\mathrm{k}} \in \mathbb{Z}_+$, $\alpha > 0$

$k \leftarrow 0$; $\boldsymbol{\lambda}_{\mathrm{L},i}(k) = \boldsymbol{\lambda}_i(k) = \mathbf{0}$, Initialize $\boldsymbol{\beta}_i(k)$, $\forall i$
Each agent i initializes $\boldsymbol{\xi}_i(k)$ by (9) given $\boldsymbol{\beta}_i(k)$
Each agent i obtains $\boldsymbol{\beta}_r(k)$ from neighbors $r \in \mathcal{N}_i$
while $k < N_{\mathrm{k}}$ **do**
  parfor agent $i = 1$ *to* N_a **do**
    Obtain $q_{i,j}(\cdot), \forall j = 1, \cdots, n$ by Algorithm 1
    Obtain $\boldsymbol{\lambda}_{\mathrm{L},r}(k)$ from neighbors $r \in \mathcal{N}_i$
    $\boldsymbol{\beta}_i(k+1) \leftarrow$ update by (15a), (15b)
    $\boldsymbol{\xi}_i(k+1) \leftarrow$ solve (9) given $\boldsymbol{\beta}_i(k+1)$
    Obtain $\boldsymbol{\beta}_r(k+1)$ from neighbors $r \in \mathcal{N}_i$
    $\boldsymbol{\lambda}_{\mathrm{L},i}(k+1), \boldsymbol{\lambda}_i(k+1) \leftarrow$ update by (15)
  $k \leftarrow k+1$

V. Numerical Results

This section presents a numerical simulation of the distributed co-design of 6-DOF manipulators. Suppose there is a complete graph with 8 agents, where each agent includes a trajectory planner (9) with its task specification. For agent-i, the cost function J_i of (9a) represents the copper loss of all n motors and is given by $J_i \triangleq \int_0^{t_{\mathrm{f},i}} \sum_{j=1}^n (i_{\mathrm{d},j}(t)^2 + i_{\mathrm{q},j}(t)^2)dt$, where $t_{\mathrm{f},i}$ varies from 2.0 to 3.5 s. For simplicity, each agent's task is to move a solid iron ball to a desired final position. Each agent's dynamics (9b) can be computed by Remark 1 with the ball mass varying from 0.5 to 1.2 kg. The initial state $\boldsymbol{x}_i(0)$ in (9c) is given by $\mathrm{col}\{\boldsymbol{\theta}_0, \mathbf{0}_{3n}\}$ with $\boldsymbol{\theta}_0 = \mathrm{col}\{-0.74, 1.06, 0.96, 2.02, 0.84, -2.19\}$ rad. The desired final position $\boldsymbol{\theta}_{\mathrm{des},i}$ in (9e) is given by $\mathrm{col}\{-3.14, 0.59, 1.79, 0, -0.81, 0\}$ rad. The joint constraints (5) are given by $\bar{\boldsymbol{\theta}} := \mathrm{col}\{2\pi, 0.6\pi, 0.6\pi, 2\pi, 0.6\pi, 2\pi\} = -\underline{\boldsymbol{\theta}}$ and $\bar{\dot{\boldsymbol{\theta}}} := 100\pi \cdot \mathbf{1}_n = -\underline{\dot{\boldsymbol{\theta}}}$, for all agents and joints. The trajectory planning problem (9) is solved by the direct collocation method and referred to [25, Section III.B] for details. Let $\rho_1 = 0.75$, $\rho_2 = 10$, $\alpha = 0.08$, $N_\tau = N_\omega = 20$.

The distributed co-design algorithm, i.e. Algorithm 2, is utilized to update $\boldsymbol{\beta}_i$, where the initial design $\boldsymbol{\beta}_i(0)$ is obtained empirically given each task. The global average of motor loss, i.e. $\sum_{i=1}^{N_a} \ell_i(\boldsymbol{\beta}_i(k))/(N_a n) \cdot 100\%$, over iterations is shown in Fig. 3. It shows that the average motor loss decreases by 4% after 60 iterations. Fig. 3 also shows the consensus error, i.e. $\sum_{(i,j)\in\mathcal{E}} ||\boldsymbol{\beta}_i(k) - \boldsymbol{\beta}_j(k)||_2$, converges to zero. It converges three times within 60 iterations because there exist multiple consensual $\boldsymbol{\beta}_i$ that satisfy all the constraints. The gradient of $\sum_{i=1}^{N_a} \ell_i(\cdot)$ and of the constraints drive $\boldsymbol{\beta}_i$ to another consensual point with a lower loss. The global average of loss does not monotonically decrease because of the non-convexity of (12). And there might not be a global optimum that is also optimal for every individual loss function. Hence, the global average increases when the consensus error decreases. Ultimately, the proposed algorithm drives $\boldsymbol{\beta}_i$ to a local and consensual optimum. Fig. 4 further validates that the distributed co-design improves

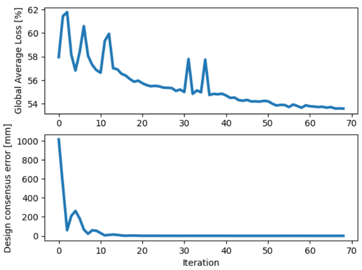

Fig. 3. The global average loss and consensus error over iterations

the motor operation efficiency. The black circles represent the motor operation points from the optimal trajectory given a particular design, where the circle radius indicates the probability of this operation point. The numerical results altogether verify the effectiveness of the proposed distributed co-design algorithm.

VI. Conclusion

This paper investigates a motion and motor co-design problem for robotic manipulators given multiple tasks. Existing co-design methodologies typically solve trajectory planning given each task individually and then optimize motor efficiency given these trajectories. With the number of tasks increasing, the computational burden grows significantly. To reduce computational burden and improve scalability as the number of tasks grows, this paper introduces a distributed co-design framework to handle the co-design process for all tasks in a distributed fashion. This paper further presents a consensus-based distributed optimization algorithm, which secures a unified set of design parameters for all tasks ultimately such that the total average of motor operation efficiency is optimized and the design constraints are satisfied across all tasks and motors. The distributed manner reduces the computational load by allowing each agent to solve co-design optimization solely for its designated task. A numerical simulation further verifies the proposed algorithm.

Future improvements include a thorough theoretical analysis of the proposed algorithm and developing a more efficient gradient descent algorithm, such as momentum-based gradient descent, etc. In addition, this paper focuses on optimizing motor efficiency based on open-loop optimal trajectories from trajectory planners. In practice, one also needs to design a closed-loop tracking controller to track the optimal trajectory for each task. Thus, how to co-design the motors, motions, and closed-loop controller of robotic manipulators given multiple tasks is a valuable direction.

References

[1] L. Zhou and S. Bai, "A new approach to design of a lightweight anthropomorphic arm for service applications," *Journal of Mechanisms and Robotics*, vol. 7, no. 3, p. 031001, 2015.

[2] O. Taylor and A. Rodriguez, "Optimal shape and motion planning for dynamic planar manipulation," *Autonomous Robots*, vol. 43, pp. 327–344, 2019.

[3] J.-T. Lin, C. Girerd, J. Yan, J. T. Hwang, and T. K. Morimoto, "A generalized framework for concentric tube robot design using gradient-based optimization," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3774–3791, 2022.

[4] G. Riva, S. Radrizzani, G. Panzani, M. Corno, and S. M. Savaresi, "An optimal battery sizing co-design approach for electric racing cars," *IEEE Control Systems Letters*, vol. 6, pp. 3074–3079, 2022.

[5] R. J. Webster III and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1661–1683, 2010.

[6] S. Rezazadeh and J. W. Hurst, "On the optimal selection of motors and transmissions for electromechanical and robotic systems," in *2014 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2014, pp. 4605–4611.

[7] M. Pettersson and J. Ölvander, "Drive train optimization for industrial robots," *IEEE Transactions on Robotics*, vol. 25, no. 6, pp. 1419–1424, 2009.

[8] Y.-S. Wang and Y. Wang, "A gradient-based approach for optimal plant controller co-design," in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 3249–3254.

[9] D. Findlay, M. Jafarinasab, and S. Sirouspour, "Optimization-based design of a novel hybrid aerial/ground mobile manipulator," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 2467–2472.

[10] G. Bravo-Palacios, A. Del Prete, and P. M. Wensing, "One robot for many tasks: Versatile co-design through stochastic programming," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1680–1687, 2020.

[11] S. Števo, I. Sekaj, and M. Dekan, "Optimization of robotic arm trajectory using genetic algorithm," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 1748–1753, 2014.

[12] L. Carlone and C. Pinciroli, "Robot co-design: beyond the monotone case," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3024–3030.

[13] M. Toussaint, J.-S. Ha, and O. S. Oguz, "Co-optimizing robot, environment, and tool design via joint manipulation planning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6600–6606.

[14] S. Ha, S. Coros, A. Alspach, J. Kim, and K. Yamane, "Computational co-optimization of design parameters and motion trajectories for robotic systems," *The International Journal of Robotics Research*, vol. 37, no. 13-14, pp. 1521–1536, 2018.

[15] M. Geilinger, R. Poranne, R. Desai, B. Thomaszewski, and S. Coros, "Skaterbots: Optimization-based design and motion synthesis for robotic creatures with legs and wheels," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–12, 2018.

[16] T. Dinev, C. Mastalli, V. Ivan, S. Tonneau, and S. Vijayakumar, "A versatile co-design approach for dynamic legged robots," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 10 343–10 349.

[17] T.-H. Wang, P. Ma, A. E. Spielberg, Z. Xian, H. Zhang, J. B. Tenenbaum, D. Rus, and C. Gan, "Softzoo: A soft robot co-design benchmark for locomotion in diverse environments," in *The Eleventh International Conference on Learning Representations*, 2022.

[18] G. Bravo-Palacios, G. Grandesso, A. D. Prete, and P. M. Wensing, "Robust co-design: Coupling morphology and feedback design through stochastic programming," *Journal of Dynamic Systems, Measurement, and Control*, vol. 144, no. 2, p. 021007, 2022.

[19] A. Spielberg, B. Araki, C. Sung, R. Tedrake, and D. Rus, "Functional co-optimization of articulated robots," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5035–5042.

[20] G. Fadini, T. Flayols, A. Del Prete, N. Mansard, and P. Souères, "Computational design of energy-efficient legged robots: Optimizing for size and actuators," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 9898–9904.

[21] Z. Lu, W. Jin, S. Mou, and B. D. Anderson, "Cooperative tuning of multi-agent optimal control systems," in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 571–576.

[22] J. Lemmens, P. Vanassche, and J. Driesen, "Pmsm drive current and voltage limiting as a constraint optimal control problem," *IEEE Journal of emerging and selected topics in power electronics*, vol. 3, no. 2, pp. 326–338, 2014.

[23] K. M. Lynch and F. C. Park, *Modern robotics*. Cambridge University Press, 2017.

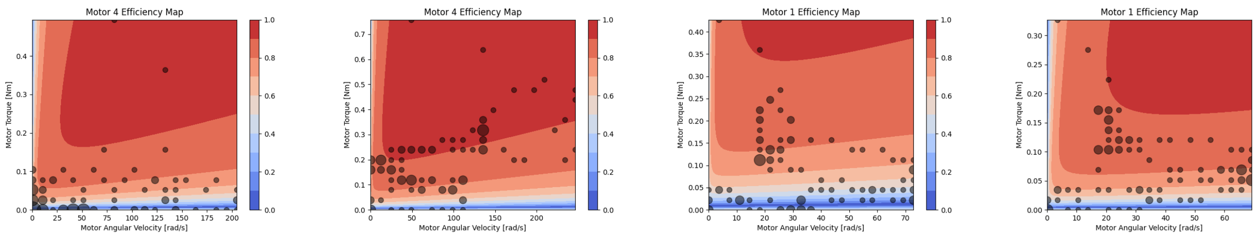

Fig. 4. The motor efficiency map of agent 1's motor 4 before and after optimization, agent 2's motor 1 before and after optimization, from left to right.

[24] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2014.

[25] A. Stein, Y. Wang, Y. Sakamoto, B. Wang, and H. Fang, "Application-oriented co-design of motors and motions for a 6dof robot manipulator," *arXiv preprint arXiv:2310.03132*, 2023.

[26] C. Godsil and G. F. Royle, *Algebraic graph theory*. Springer Science & Business Media, 2001, vol. 207.

[27] Z. Lu and S. Mou, "A distributed algorithm for multi-agent optimization under edge-agreements," *arXiv preprint arXiv:2305.17240*, 2023.

[28] J. Giesen and S. Laue, "Combining admm and the augmented lagrangian method for efficiently handling many constraints." in *IJCAI*, 2019, pp. 4525–4531.

[29] T. Higuchi, T. Abe, and Y. Yokoi, *PRINCIPLE AND DESIGN OF AC MOTORS (in Japanese)*. Kagakujohoshuppan, 2017.

APPENDIX

A. Magnetic Equivalent Circuit Modeling for SPMSM

The magnetic equivalent circuit (MEC) modeling for an arbitrary SPMSM [29] is presented in this subsection for completeness. The SPMSM design variables are summarized in Table I. This appendix omits the index i. The index j indicates the j-th motor of an arbitrary manipulator. $\gcd(a, b)$ denotes the greatest common divisor of two positive integers a and b. The constant parameters for an SPMSM are:

- Number of slots $Q = 12$
- Number of slots per phase $q_1 = Q/3$
- Number of pole pairs $p = 4$
- Number of slots per pole per phase $q_{\text{pm}} = \frac{q_1}{\gcd(q_1, 2p)}$
- Number of winding turns per tooth $n_{\text{s}} = 50$
- Number of coils connected in parallel $C_{\text{p}} = 1$
- Gear ratio at the j-th joint $Z_j = 50\ \forall j$
- Height of tooth tip $h_{\text{tip}} = 2$ mm
- Width of air gap $\delta = 0.5$ mm
- Magnet width in electric angle $\alpha_{\text{m}} = \pi$
- Remanent flux density of the magnet $B_{\text{r}} = 1.38$ T
- Maximum limitation for flux density $B_{\max} = 1.5$ T
- Mass density of iron $\rho_{\text{iron}} = 7.8 \cdot 10^{-6}$ kg/mm^3
- Mass density of copper $\rho_{\text{cu}} = 8.93 \cdot 10^{-6}$ kg/mm^3
- Electric resistivity of copper winding $\rho_{\text{e}} = 1.8 \cdot 10^{-5}$ Ω·mm
- Permeability of air $\mu_0 = 4\pi \cdot 10^{-7}$ N/A^2
- Relative recoil permeability of the magnet $\mu_{\text{r}} = 1.05$
- Filling factor $f_{\text{f}} = 0.55$

1) Geometric Parameters: According to Fig. 2, the expression for the slot height is

$$h_{\text{ss},j} = r_{\text{so},j} - h_{\text{sy},j} - r_{\text{ro},j} - \delta - h_{\text{tip}}. \tag{16}$$

For a rectangular tooth cross-section, one can compute the slot width as $b_{\text{ss},j} = A_{\text{slot},j}/h_{\text{ss},j}$, where $A_{\text{slot},j}$ is the slot area, i.e.

$$A_{\text{slot},j} = \frac{\pi((r_{\text{so},j} - h_{\text{sy},j})^2 - (r_{\text{ro},j} + \delta + h_{\text{tip},j})^2)}{Q} - w_{\text{tooth},j} h_{\text{ss},j}. \tag{17}$$

The cross-section area of the stator core is given by

$$A_{\text{so},j} = \pi r_{\text{so},j}^2 - \pi(r_{\text{ro},j} + \delta)^2 - Q(A_{\text{slot},j} + b_0 h_{\text{tip}}). \tag{18}$$

Thus the volume of the stator core is given by

$$V_j = A_{\text{so},j} l_j. \tag{19}$$

The copper area is given by $A_{\text{cu},j} = A_{\text{slot},j} f_f$. For concentrated windings and assuming one winding is a complete turn around a tooth, the area of a single coil is given by $A_{\text{coil},j} = A_{\text{cu},j}/(2n_{\text{s}})$. The minimal wire diameter is

$$D_{\text{wire},j} = \sqrt{4A_{\text{coil},j}/\pi}. \tag{20}$$

The arc span per slot can be determined by $\tau_{\text{s},j} = 2\pi(r_{\text{ro},j} + \delta)/Q$. The average length of the coil end-winding $l_{\text{end,av},j}$ and the total coil length $l_{\text{coil},j}$ are given by

$$\begin{aligned} l_{\text{end,av},j} &= (w_{\text{tooth},j}(2 - \pi/2) + \pi\tau_{\text{s},j}/2)/2, \\ l_{\text{coil},j} &= 2l_j + 2l_{\text{end,av},j}. \end{aligned}$$

Then the weight of the stator and rotor are given by:

$$\begin{aligned} m_{\text{rotor},j} &= \rho_{\text{iron}} \pi r_{\text{ro},j}^2 l_j, \\ m_{\text{stator},j} &= \rho_{\text{iron}} \pi r_{\text{so},j}^2 l_j - \rho_{\text{iron}} \pi (r_{\text{ro},j} + \delta)^2 l_j \\ &\quad - \rho_{\text{iron}} A_{\text{slot},j} l_j Q + \rho_{\text{cu}} A_{\text{coil},j} l_{\text{coil},j} n_{\text{s}} Q. \end{aligned} \tag{21}$$

Without loss of generality, define the x-axis as the central axis of each rotor or stator, i.e. x-axis coincides with the axial length of core l_j; consequently, define the y-axis and z-axis by following the right-hand rule and the two axes indicate the central radius. All three axes originate at the centroid of the rotor or stator, i.e. the center of the axial length l_j. Since each rotor is a solid cylinder, the moment of inertia about three principal axes of each rotor is given by:

$$\begin{aligned} I_{\text{xx},j} &= \tfrac{1}{2}\rho_{\text{iron}} \pi r_{\text{ro},j}^4 l_j = \tfrac{1}{2} m_{\text{rotor},j} r_{\text{ro},j}^2, \\ I_{\text{yy},j} &= I_{\text{zz},j} = \tfrac{1}{12}\rho_{\text{iron}} \pi r_{\text{ro},j}^2 l_j (3r_{ro,j}^2 + l_j^2). \end{aligned}$$

To simplify the inertia calculation for stators, each stator is simplified as a hollow cylinder with outer radius $r_{\text{so},j}$ and inner radius $r_{\text{ro},j} + \delta$. Then the moment of inertia about three principal axes of each stator is given by:

$$\begin{aligned} I_{\text{xx},j} &= \tfrac{1}{2} m_{\text{stator},j} (r_{\text{so},j}^2 + (r_{\text{ro},j} + \delta)^2), \\ I_{\text{yy},j} &= I_{\text{zz},j} = \tfrac{1}{12} m_{\text{stator},j} (3(r_{\text{so},j}^2 + (r_{\text{ro},j} + \delta)^2) + l_j^2). \end{aligned}$$

2) Resistance: The resistance per tooth is given by $R_{1,j} = (n_s^2 \rho_e l_{\text{coil},j})/(A_{\text{slot},j} f_f)$. Using this, the phase resistance can be calculated as

$$R_j = q_1 R_{1,j}/C_p^2. \tag{22}$$

3) Permeance: The permeance of the magnetic path across the air gap and the slot opening, denoted by $p_{g,j}$ and $p_{\text{so},j}$, are given by:

$$p_{g,j} = \frac{2\pi r_{\text{ro},j}\mu_0 l/Q}{\delta + h_{\text{m},j}/\mu_r}, \quad p_{\text{so},j} = \frac{\mu_0 h_{\text{tip}} l_j}{b_{0,j}}.$$

The permeance of the magnetic path that curves from tip to tip is given by $p_{\text{tt},j} = \frac{\mu_0(\delta + h_{\text{m},j}) l_j}{\pi(\delta + h_{\text{m},j})/2 + b_{0,j}}$.

4) Inductance: For an SPMSM, its d-axis and q-axis inductance are equivalent to each other and given by

$$L_{\text{d},j} = L_{\text{q},j} = q_1 n_s^2 L_{1,j}/C_p^2, \tag{23}$$

where $L_{1,j}$ is the inductance per turn and per tooth, given by $L_{1,j} = p_{g,j} + 3p_{\text{so},j} + 3p_{\text{tt},j}$.

5) Flux: To proceed with the calculation, it is necessary to determine Carter's coefficient denoted by $k_{C,j}$, given by

$$k_{C,j} = \frac{t_{\text{pitch},j}}{t_{\text{pitch},j} - \gamma_j \delta}, \ \gamma_j = \frac{(b_{0,j}/\delta)^2}{5 + b_{0,j}/\delta}, \ t_{\text{pitch},j} = \frac{2\pi r_{\text{ro},j}}{Q}. \tag{24}$$

Then, the magnetic flux density across the gap is given by $B_{g,j} = B_r \frac{h_{\text{m},j}/\mu_r}{h_{\text{m},j}/\mu_r + \delta k_{C,j}}$. The flux density corresponding to the first harmonics can be calculated as $B_{g,1,j} = 4B_{g,j}/\pi$. Consequently, the flux per tooth per single turn is given by

$$\Phi_{1,j} = B_{g,1,j} l_j 2\pi r_{\text{ro},j}/Q. \tag{25}$$

In the absence of skewness, the flux linkage is given by

$$\Phi_{\text{m},j} = k_w n_s \Phi_{1,j} q_1 / C_p, \tag{26}$$

where $k_w = k_p k_d$ denotes the winding factor, and

$$k_p = \sin(\pi p/Q), \quad k_d = \frac{\sin(\pi/6)}{q_{\text{pm}} \sin(\pi/(6q_{\text{pm}}))}. \tag{27}$$

B. SPMSM Torque & Efficiency Modeling

This subsection introduces the modeling of SPMSM torque, maximum torque, and efficiency. The constant parameters used here are:

- Operating temperature $T_{\text{op}} = 80$ °C
- Ambient temperature $T_{\text{amb}} = 20$ °C
- Motor's q-axis voltage upper bound $\overline{u}_{q,j} = 100$ V
- Motor's q-axis current upper bound $\overline{i}_{q,j} = 3$ A
- Maximum motor power $P_{\max,j} = 600$ W

1) Torque: First, the Hysteresis loss and eddy current loss are given by Steinmetz's equation, i.e.

$$P_{\text{hyst},j} = k_{\text{hyst}} \left|\frac{p\omega_j}{2\pi}\right| B_j^{1.6} V_j, \tag{28a}$$

$$P_{\text{eddy},j} = k_{\text{eddy}} \left(\frac{p\omega_j}{2\pi}\right)^2 B_j^2 V_j, \tag{28b}$$

where $k_{\text{hyst}} = 130$ WsT$^{-1.6}$m^{-3} and $k_{\text{eddy}} = 1.1$ Ws^{2}T^{-2}m^{-3} are the coefficients, estimated by finite element analysis (FEA) simulation of a base design motor. V_j is the volume of motor-j's stator core and defined in (19); B_j is the average flux density at the tooth, calculated as follows:

$$B_j = \frac{\Phi_j}{n_s (q_1/C_p) w_{\text{tooth},j} l_j}, \tag{29a}$$

$$\Phi_j \triangleq \sqrt{(\Phi_{\text{m},j} + L_{\text{d},j} i_{\text{d},j})^2 + (L_{\text{q},j} i_{\text{q},j})^2}, \tag{29b}$$

where Φ_j indicates the total flux generated by both permanent magnets and coils. The loss torque is given by

$$\tau_{\text{hyst},j} = P_{\text{hyst},j}/|\omega_j|, \ \tau_{\text{eddy},j} = P_{\text{eddy},j}/|\omega_j|. \tag{30}$$

Then the motor torque $\tau_{\text{m},j}$ of motor j is given by

$$\hat{\tau}_{\text{m},j} \triangleq 1.5p\Phi_{\text{m},j} i_{\text{q},j} + (L_{\text{d},j} - L_{\text{q},j}) i_{\text{d},j} i_{\text{q},j}, \tag{31a}$$

$$\tau_{\text{m},j} = \hat{\tau}_{\text{m},j} - \text{sign}(\hat{\tau}_{\text{m},j})(\tau_{\text{hyst},j} + \tau_{\text{eddy},j}), \tag{31b}$$

where $\text{sign}(\cdot)$ denotes the signum function. Then the joint torque τ_j at joint j is given by

$$\omega_j := \dot{\theta}_j Z_j, \ \tau_j = \tau_{\text{m},j} Z_j, \tag{32}$$

where $\dot{\theta}_j$ is the angular velocity of joint j.

Given an arbitrary optimal trajectory of joint-j's position and velocity and motor-j's current and voltage from a trajectory planner (9), the motor torque and velocity at each time instance can be calculated by (31) and (32), respectively. Then the maximum (magnitude) motor torque $\tau_{\max,j}$ and the maximum (magnitude) motor velocity $\omega_{\max,j}$ of motor j can be obtained directly from the trajectory of motor-j's torque and velocity.

2) Efficiency: The efficiency $\eta_j(\tau_{\text{m},j}, \omega_j, \boldsymbol{\beta}_j)$ is determined by motor-j's torque $\tau_{\text{m},j}$ and motor's velocity ω_j under the motor design parameter $\boldsymbol{\beta}_j \in \mathbb{R}^7$. The efficiency $\eta_j(\tau_{\text{m},j}, \omega_j, \boldsymbol{\beta}_j)$ is given by:

$$\eta_j = (P_{\text{in},j} - P_{\text{cu},j} - \hat{P}_{\text{hyst},j} - \hat{P}_{\text{eddy},j})/P_{\text{in},j}, \tag{33}$$

where $P_{\text{in},j}, P_{\text{cu},j}$ are the input power and copper losses power of motor j, respectively. $P_{\text{in},j}, P_{\text{cu},j}$ are given by

$$P_{\text{in},j} = u_{\text{d},j} i_{\text{d},j} + u_{\text{q},j} i_{\text{q},j}, P_{\text{cu},j} = R_{\text{T},j}(i_{\text{d},j}^2 + i_{\text{q},j}^2), \tag{34a}$$

$$R_{\text{T},j} = R_j T/T_{\text{amb}}, \ T = T_{\text{op}} P_{\text{in},j}/P_{\max,j} + T_{\text{amb}}, \tag{34b}$$

where R_j is defined in (22); T is the scaled operating temperature; (34b) calculates the copper resistance given scaled temperature by linear extrapolation. $\hat{P}_{\text{hyst},j}$, $\hat{P}_{\text{eddy},j}$, currents and voltages are calculated as follows:

$$\Phi_{\max,j} \triangleq (\overline{u}_{\text{q},j} - R_j \frac{T_{\text{op}}}{T_{\text{amb}}} \overline{i}_{\text{q},j})/(p|\omega_j|), \tag{35a}$$

$$\hat{\Phi}_j \triangleq \min(\Phi_{\max,j}, \Phi_{\text{m},j}), \tag{35b}$$

$$i_{\text{d},j} = (\hat{\Phi}_j - \Phi_{\text{m},j})/L_{\text{d},j}, \tag{35c}$$

$$i_{\text{q},j} = (|\tau_{\text{m},j}| + \hat{\tau}_{\text{hyst},j} + \hat{\tau}_{\text{eddy},j})/(1.5p\Phi_{\text{m},j}), \tag{35d}$$

$$u_{\text{d},j} = R_j i_{\text{d},j} - p|\omega_j| i_{\text{q},j} L_{\text{d},j}, \tag{35e}$$

$$u_{\text{q},j} = R_j i_{\text{q},j} + p|\omega_j| \hat{\Phi}_j, \tag{35f}$$

where $\Phi_{\max,j}$ is the maximum flux given motor velocity and maximum voltage and current; (35b) ensures flux weakening; $\hat{P}_{\text{hyst},j}$, $\hat{P}_{\text{eddy},j}$, $\hat{\tau}_{\text{hyst},j}$, $\hat{\tau}_{\text{eddy},j}$ are calculated by following (28), (29a), and (30), where Φ_j in (29a) is replaced by $\hat{\Phi}_j$ in (35b).