Tri-Space Operational Control of Redundant Multilink and Hybrid Cable-Driven Parallel Robots Using an Iterative-Learning-Based Reactive Approach

Dipankar Bhattacharya, *Member, IEEE*, Yin Pok Chan, Siqi Shang, Yuen Shan Chan, Ying Tan, *Fellow, IEEE*, and Darwin Lau, *Senior Member, IEEE*

***Abstract*— Cable-driven parallel robots (CDPRs) are a type of parallel mechanism in which cables are used as actuators. Due to the two levels of redundancy and numerous constraints within the CDPR actuation, joint and operational spaces (together known as the tri-space), tracking a given trajectory in the operational space while satisfying constraints in tri-space simultaneously is challenging. To the best of the authors' knowledge, there does not exist any tri-space control framework, which is robust, effective, and directly applicable to several architectures of redundantly actuated CDPRs. This article proposes a tri-space control framework that combines reactive control (RC) and iterative-learning control (ILC) to perform repetitive tasks in the operational space. The framework allows the tracking of operational space trajectories online with feasible cable forces while avoiding undesirable situations such as cable-link interference, joint interference, and loss of manipulability. On the other hand, by finding an optimal parameter in the null space using a novel parameterization of a null-space vector, the performance can be improved through ILC when the task is repeatedly executed. Simulation and hardware results on various multilink cable-driven robots (MCDRs) and hybrid cable-driven robots (HCDRs) show that the proposed tri-space control framework can be conveniently and effectively applied to the real-time control of different CDPRs.**

***Index Terms*— Cable-driven robots, iterative-learning control, reactive control (RC), tri-space control.**

I. Introduction

CABLE-DRIVEN parallel robot (CDPR) are a type of parallel mechanism actuated by cables instead of rigid links, having advantages of low inertia, large potential workspace, high reconfigurability, and transportability. Hence, CDPRs have been widely studied and applied in manufacturing [1], [2], building construction [3], [4], rehabilitation [5], and rapid prototyping [6]. However, the control of CDPRs is difficult due to the unique property that cables can only apply pulling forces (positive cable forces), resulting in a redundantly actuated system. This redundancy generates challenges in their mechanical design [7], workspace analysis [8], [9], [10], [11], [12], and control and synthesis [13], [14], [15], [16].

Single-link cable-driven robots (SCDRs), a common type of CDPR, consist of a single rigid body end-effector actuated by cables connected from the base frame. However, this design requires the base frame to completely encapsulate the desired workspace, leading to a large robot footprint. Moreover, SCDRs typically have a limited range of end-effector orientations. Multilink cable-driven robots MCDRs and hybrid cable-driven robots (HCDRs) have been developed to relax such limitations.

MCDRs are a type of CDPR consisting of multiple rigid links in order to combine the dexterity and increased end-effector orientation capability of serial manipulators with the actuation advantages of CDPRs. In addition, the similarities between MCDRs and anthropomorphic systems have motivated biorobotic applications, such as musculoskeletal arms [17], humanoid [18] robots, and snake-like robots [19]. HCDRs are hybrid actuated systems that combine cables and direct joint actuation and possess both passive and active joint degrees of freedoms (DoFs). Examples of HCDRs include: 1) the FASTKIT robot [20], where a CDPR is mounted onto multiple mobile robot platforms such that these mobile robots can enlarge their workspace and reconfigure the attachment points, and 2) the SpiderArm robot, where robot arms are mounted onto the end-effector of an SCDR to increase the dexterity and accuracy of the CDPR while still maintaining a large translational workspace. In this work, CDPRs will be referred to as MCDRs and HCDRs.

Compared to SCDRs, the control of CDPRs is more complicated since the kinematics and dynamics have to be described within three spaces: 1) actuation space refers to the control input to the robot, such as cable forces and length commands

Manuscript received 3 January 2023; revised 23 February 2023; accepted 16 March 2023. Date of publication 13 April 2023; date of current version 23 October 2023. This work was supported in part by the Research Grants Council (General Research Fund) under Grant 14203921 and in part by the Innovation and Technology Commission (University–Industry Collaboration Programme) under Grant UIM/345. Recommended by Associate Editor T. Hatanaka. *(Corresponding author: Dipankar Bhattacharya.)*

Dipankar Bhattacharya, Yin Pok Chan, Siqi Shang, Yuen Shan Chan, and Darwin Lau are with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong, SAR (e-mail: dipankarbhattacharya@cuhk.edu.hk; ypchan@cuhk.edu.hk; sqshang@link.cuhk.edu.hk; yschan03@cuhk.edu.hk; darwinlau@mae.cuhk.edu.hk).

Ying Tan is with the Faculty of Engineering and Information Technology, The University of Melbourne, Parkville, VIC 3010, Australia (e-mail: yingt@unimelb.edu.au).

Video link for the paper: https://www.youtube.com/watch?v=k8JaL80ZBbQ

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCST.2023.3263689.

Digital Object Identifier 10.1109/TCST.2023.3263689

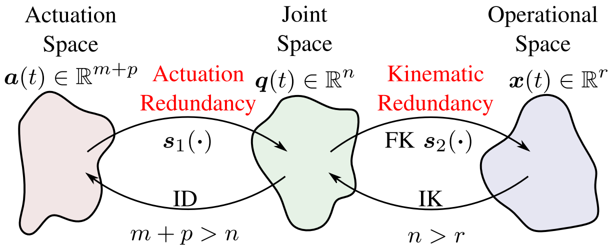

Actuation Space
$\boldsymbol{a}(t) \in \mathbb{R}^{m+p}$
Actuation Redundancy
Joint Space
$\boldsymbol{q}(t) \in \mathbb{R}^{n}$
Kinematic Redundancy
Operational Space
$\boldsymbol{x}(t) \in \mathbb{R}^{r}$
$\boldsymbol{s}_1(\cdot)$
FK $\boldsymbol{s}_2(\cdot)$
ID
IK
$m+p>n$
$n>r$

Fig. 1. Tri-space mapping (shown by arrows) between the actuation, joint, and operational spaces of MCDRs and HCDRs for $m+p$ actuators, n-dimensional joint space, and r-dimensional operational space such that $\boldsymbol{s}_1 : \boldsymbol{a} \to \boldsymbol{q}$ and $\boldsymbol{s}_2 : \boldsymbol{q} \to \boldsymbol{r}$. In this figure, ID means inverse dynamics, FK means forward kinematics, and IK means inverse kinematics.

for the actuated DoF with $m + p$ DoF; 2) joint space represents the DoF (generalized coordinates) with n DoF; and the 3) operational space (or task space) denotes the pose of the end-effector with the DoF of r. The combination of actuation, joint, and operational spaces is defined as the tri-space. The dimensions in the tri-space usually satisfy $r < n < m + p$, as shown in Fig. 1. For example, in the SpiderArm robot, $r = 6$, $n = 12$, and $m + p = 8 + 6 = 14$.

Due to the tri-space setting in CDPRs, the control of CDPRs has two levels of redundancies. One is between operational and joint spaces (kinematic redundancy) and the other is between joint and cable actuation spaces (actuation redundancy). The kinematic redundancy leads to potentially infinite joint poses that result in the same operational space motion and the actuation redundancy leads to potentially infinite sets of actuation commands to produce the same joint motion. The infinitely many solutions from redundancies always conflict with various constraints, such as positive cable forces, joint limits, and cable interference.

Four requirements are identified in developing the tri-space control framework. First, the framework can solve the tri-space redundancy problem in the CDPRs by simultaneously determining actuation and joint space motion variables in a single process with the consideration of tri-space constraints. Second, the avoidance functions in the framework can guide the CDPRs to smoothly avoid undesirable situations such as cable-link interference, loss of manipulability, and reaching mechanical joint limits in advance. Third, it can improve the performance if the operational task repeats itself. Finally, this framework should provide a feasible solution with a relatively lower computational cost. Consequently, this work proposes a generalized tri-space operational control framework by combining reactive control (RC) and iterative-learning control (ILC) for the CDPRs.

Direct reactive operational space control or RC approaches have been used to directly determine a set of feasible actuation commands at each time step from an operational space trajectory [21]. However, a drawback for RC is that it only considers the instantaneous pose and does not consider evolution of the pose and control input. Typically, the task performance is based on the entire trajectory, where reactive methods struggle. Besides, existing RC approaches [22] are unable to handle constraints, such as lack of manipulability [23], joint limits [24], or cable interference [25]. ILC is well-known for its ability to improve the tracking performance when the task repeats itself [26], [27], [28], [29]. Traditionally, ILC updates control actions at each sampling instant within the given time interval with appropriate convergence conditions to guarantee convergence [26], [27].

Hence, in this work, RC is used to formulate the reference tracking in the operational space with the consideration of constraints in the tri-space. The resultant problem is then transformed into an optimization problem with an unknown parameter set to balance the control effort and avoidance of the constraints. In addition, the projection of the joint space acceleration onto its null space is parameterized by another set of unknown parameters. The role of ILC is thus to learn two sets of unknown parameters to improve the performance when the task is repetitive. By carefully choosing initial values of parameters in these two sets, the feasible solutions of control of CDPRs are guaranteed. In order to reduce the ILC computational cost, this work utilizes novel two set parameterizations. One set balances the control effort and the satisfaction of constraints. The other set is the best parameterization in its null space. Moreover, the well-known optimization techniques, such as pattern search (PS) and Particle Swarm Optimization (PSO), are used to find optimal parameters iteratively.

The main contributions of this work are summarized as follows.

1) A novel CDPR tri-space operational control framework is proposed. This framework can track the reference trajectory in task space, handle kinematic and actuation redundancy, avoid cable-link interference, prevent loss of manipulability, and avoid reaching mechanical joint limits simultaneously. To the best of the authors' knowledge, it is the first time that such a control framework is developed to handle multilevel redundancies and various constraints for CDPR.
2) The proposed framework combines RC, which provides feasible and optimal performance in the operational space and constraints in tri-space, with ILC to find two sets of optimal parameters. One set tries to balance the control effort and satisfy constraints, while the other set is the parameterization in the null space. The proposed framework is thus able to improve the task performance over iterations with a low computational cost.
3) The capabilities of the control framework are illustrated through simulations of various operational space trajectories using the Cable-Robot Analysis and Simulation Platform for Research (CASPR) software[1] [30] on a biologically inspired mechanical arm (BMArm) robot (two-link MCDR), and the SpiderArm and FASTKIT-Planar HCDRs. Experiments on the BMArm are also conducted to show the robustness of the proposed framework.

The remainder of this article is organized as follows. Section II discusses the related work. Section III provides important notations, generalized MCDR and HCDR models

[1] All methods and algorithms in this work have been implemented in the open-source cable robot software CASPR and can be accessed from: http://www.github.com/darwinlau/CASPR

with examples, and control objective. Section IV discusses multiple CDPR control challenges and outlines the tri-space operational control framework. Sections V and VI propose the reactive and ILC components, respectively. Section VII presents the control framework stability analysis. Simulation and hardware results are presented in Sections VIII and IX, respectively. Finally, Section X concludes this article and discusses future work.

II. Related Work

This section revisits the literature about handling redundancies of CDPRs. A lot of work has focused on SCDR, which only involved a redundancy between joint and operational space. For example, in [31], a dual-space control scheme was proposed. The uncertainties in the dynamic models make an accurate feedforward control impossible. Consequently, the proposed tension distributions algorithm cannot guarantee a feasible solution within the entire robot workspace. To deal with disturbances appearing in the un-actuated joints, Qi et al. [32] proposed a minimum actuated joint torque control for the actuated joints while resolving the kinematic redundancy without considering the tracking performance in joint space and the constraints.

In [33], a tricriteria optimization-coordination-motion scheme was proposed for dual-redundant robots, which combined minimum velocity norm, repetitive motion planning, and infinity-norm velocity minimization solutions into a single quadratic programming (QP) formulation. In order to solve such a QP problem, different techniques were proposed. For example, the parameter-varying-based neural networks were proposed in [34] and [35]. In [36], the training-free NN-based technique was proposed to deal with both convex and nonconvex sets while removing the initial error and accumulated error issues for redundant robots. In [37], a mutual collision avoidance scheme was proposed by using a line segment-based distance measurement algorithm to solve the motion planning problem of dual manipulators. The above techniques have been shown to be efficient and accurate for solving a dual-redundancy problem. However, how to balance the tracking accuracy, control effort, and satisfaction of constraints has not been discussed. This is important otherwise tracking becomes unstable for some joint configuration and eventually diverges (see Section VIII-A).

Another common approach is the cascading method (or two-stage optimization). The first stage determines a joint space trajectory that can track the reference by solving a serial robot inverse kinematics (IK) problem. The second stage is to compute the cable forces by solving the CDPR inverse dynamics (ID) problem [14], [38]. As the two stages are independent of each other, there might be some conflicts between them, i.e., the reference trajectory may not be feasible in the actuation space as pointed out in [21]. In [39], decoupled horizontal and vertical vibration models were developed, leading to a model predictive control of an HCDR. Such a technique can reduce trajectory tracking errors and vibrations without exploring the kinematic redundancy. In [40], standard kinematic resolution techniques based on local optimization of the null-space component were used to generate feasible generalized coordinates for HCDR end-effector tracking task. However, the work only focused on the kinematic redundancy and not on the actuation redundancy.

In [22], Khatib's [21] operational space control was extended to determine the muscle forces required to achieve reactive operational space control of biomechanical systems in which the workspace is conservatively approximated by a set of linear constraints [41]. As such, the control command can be determined by solving convex QP. However, for CDPRs, the linear constraints approximation of the pregenerated workspace is not practical due to the high workspace irregularity and computational costs for high DoF robots. As such, the approach proposed in [22] for biomechanical systems cannot be effectively used for CDPRs in general. Hence, in summary, there is no effective solution for a tri-space operational control to deal with two-level redundancy problem with the consideration of feasibility in the CDPRs, which motivates the proposed tri-space control framework for CDPRs.

III. Notations, CDPR Modeling, and Control Objective

This section provides relevant mathematical notations and the generalized modeling of CDPRs including MCDRs and HCDRs, followed by the control objective of this work.

A. Notations

Let $\mathbb{N}$ represent the set of all nonnegative integers. The notation $\mathbb{R}$ represents the set of all real numbers, $\mathbb{R}_{\geq 0}$ is the set of all nonnegative real numbers, and $\mathbb{R}_{>0}$ is the set of all positive real numbers. For any vector $\boldsymbol{x} \in \mathbb{R}^n$, $\|\boldsymbol{x}\|_2$ represents its Euclidean norm, defined as $\|\boldsymbol{x}\|_2 \triangleq (\boldsymbol{x}^T\boldsymbol{x})^{1/2}$, where $(\cdot)^T$ represents the transpose. For two vectors in $\mathbb{R}^n$ with $\boldsymbol{x} = [\,x_1 \ \cdots \ x_n\,]^T$ and $\boldsymbol{y} = [\,y_1 \ \cdots \ y_n\,]^T$, $\boldsymbol{x} \leq \boldsymbol{y}$ indicates that $x_i \leq y_i$, $\forall\ i = 1, \ldots, n$.

For a matrix $A = \{a_{i,j}\} \in \mathbb{R}^{n\times m}$, $\|A\|_2$ is the induced matrix norm defined as $\|A\|_2 = (A^TA)^{1/2}$. The Frobenius norm of A is defined as $\|A\|_F = (\sum_{i=1}^{n}\sum_{j=1}^{m} a_{i,j}^2)^{1/2}$. The pseudoinverse $A^\dagger$ is defined such that $AA^\dagger = I_n$, where I_n denotes an n-dimensional square identity matrix. The null space of matrix A is the set $\mathcal{N}_A = \{\mathbf{x} \in \mathbb{R}^m : A\mathbf{x} = \mathbf{0}\}$. The matrix $N_A \in \mathbb{R}^{m\times(m-n)}$ denotes a matrix whose columns form a basis of the set $\mathcal{N}_A$.

B. SCDR and MCDR Modeling

An MCDR is a type of CDPR where the number of links is more than one. If the number of links is one, then the MCDR becomes an SCDR. Let the joint space $\boldsymbol{q} = [q_1, \ldots, q_{n_M}]^T \in \mathbb{R}^{n_M}$ be defined as the generalized coordinates. The cable lengths and forces are denoted as $\boldsymbol{l} = [l_1, \ldots, l_{m_M}]^T \in \mathbb{R}^{m_M}$ and $\boldsymbol{f} = [f_1, \ldots, f_{m_M}]^T \in \mathbb{R}^{m_M}$, respectively. Hence, the kinematic relationship between $\boldsymbol{l}$ and $\boldsymbol{q}$ is described by

$$\boldsymbol{l} = \boldsymbol{s}_1(\boldsymbol{q}) \tag{1}$$

where the nonlinear mapping $\boldsymbol{s}_1 : \mathbb{R}^{n_M} \to \mathbb{R}^{m_M}$ is the IK function [42]. The derivative of (1) gives

$$\dot{\boldsymbol{l}} = \boldsymbol{L}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{2}$$

where $\boldsymbol{L}(\boldsymbol{q}) = [\partial \boldsymbol{s}_1(\boldsymbol{q})/\partial q_1, \ldots, \partial \boldsymbol{s}_1(\boldsymbol{q})/\partial q_{n_M}] \in \mathbb{R}^{m_M \times n_M}$ is the joint-cable Jacobian matrix. Due to the unilateral actuation property of cables, actuation redundancy ($m_M \geq n_M + 1$) is needed to produce motion in all DoF. The dynamics of the system is described by the following equation of motion (EoM):

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\dot{\boldsymbol{q}}, \boldsymbol{q}) + \boldsymbol{G}(\boldsymbol{q}) = -\boldsymbol{L}(\boldsymbol{q})^T \boldsymbol{f} \tag{3}$$

$$\boldsymbol{0} \leq \underline{\boldsymbol{f}} \leq \boldsymbol{f} \leq \overline{\boldsymbol{f}}. \tag{4}$$

The left-hand side of (3) refers to the system wrench, composed of the mass-inertia matrix $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n_M \times n_M}$, the centrifugal and Coriolis vector $\boldsymbol{C}(\dot{\boldsymbol{q}}, \boldsymbol{q}) \in \mathbb{R}^{n_M}$, and the gravitational vector $\boldsymbol{G}(\boldsymbol{q}) \in \mathbb{R}^{n_M}$. The joint-cable Jacobian matrix $\boldsymbol{L}(\boldsymbol{q})$ maps the cable force vector $\boldsymbol{f}$ into the joint space system wrench. Cable forces are requested to be bounded by the minimum and maximum positive force bounds $\underline{\boldsymbol{f}}$ and $\overline{\boldsymbol{f}}$ in (4), respectively.

The MCDR operational space can be defined as the position and/or orientation of the end-effector $\boldsymbol{x} = [x_1, \ldots, x_r]^T \in \mathbb{R}^r$, where r refers to the number of operational space DoF. Usually, $r \ll n_M$. For this work, Euler angles are chosen over quaternions because quaternions are much less intuitive than Euler angles. However, since the framework is generic, hence, quaternions can also be used to represent the operational space. The kinematic relationship between the joint and operational space (Euler space) can be written as

$$\boldsymbol{x} = \boldsymbol{s}_2(\boldsymbol{q}). \tag{5}$$

In (5), $\boldsymbol{s}_2 : \mathbb{R}^{n_M} \to \mathbb{R}^r$ represents the Forward Kinematics (FK), mapping the joint space to operational space $\mathbb{R}^r$. The derivative of (5) becomes

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{6}$$

where $\boldsymbol{J}(\boldsymbol{q}) = [\partial \boldsymbol{s}_2(\boldsymbol{q})/\partial q_1, \ldots, \partial \boldsymbol{s}_2(\boldsymbol{q})/\partial q_{n_M}] \in \mathbb{R}^{r \times n_M}$ is the joint-operational Jacobian matrix. This leads to the following derivative of (6):

$$\ddot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}}. \tag{7}$$

Equation (7) can be rewritten as

$$\ddot{\boldsymbol{q}} = \boldsymbol{J}^{\dagger}(\boldsymbol{q})(\ddot{\boldsymbol{x}} - \dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}}) + \ddot{\boldsymbol{q}}_N \tag{8}$$

where $\ddot{\boldsymbol{q}}_N$ and $\boldsymbol{J}^{\dagger}$ are null-space vector and pseudoinverse of $\boldsymbol{J}$, respectively.

C. HCDR Modeling

HCDR are hybrid actuated systems that combine cables and other types of rigid-link actuators. For an HCDR with m-cables with its CDPR joint space $\boldsymbol{q}_M \in \mathbb{R}^{n_M}$ and p-actuators, the original CDPR joint space $\boldsymbol{q}_M$, is extended to include the rigid-link actuated DoF $\boldsymbol{q}_a \in \mathbb{R}^p$. As a result, the joint space of the HCDR becomes $\boldsymbol{q} = [\boldsymbol{q}_M^T \;\; \boldsymbol{q}_a^T]^T \in \mathbb{R}^n$, where $n = n_M + p$. The kinematic relationship between the cable lengths $\boldsymbol{l}$ and CDPR joint space $\boldsymbol{q}_M$ is the same as that of (1) and (2) with appropriate dimensions. The actuation command $\boldsymbol{a}$ is formed by combining the cable force $\boldsymbol{f} \in \mathbb{R}^m$ and the active joint torque $\boldsymbol{\tau} \in \mathbb{R}^p$ as

$$\boldsymbol{a} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{\tau} \end{bmatrix} := \begin{bmatrix} \boldsymbol{a}_C \\ \boldsymbol{a}_D \end{bmatrix} \in \mathbb{R}^{m+p}. \tag{9}$$

The bounds on $\boldsymbol{a}$ are defined as

$$\begin{gathered} \underline{\boldsymbol{a}} \leq \boldsymbol{a} \leq \overline{\boldsymbol{a}} \\ \underline{\boldsymbol{a}} = \begin{bmatrix} \underline{\boldsymbol{f}} \\ \underline{\boldsymbol{\tau}} \end{bmatrix}, \quad \overline{\boldsymbol{a}} = \begin{bmatrix} \overline{\boldsymbol{f}} \\ \overline{\boldsymbol{\tau}} \end{bmatrix} \end{gathered} \tag{10}$$

where $\underline{\boldsymbol{\tau}}, \overline{\boldsymbol{\tau}} \in \mathbb{R}^p$ are the minimum and maximum joint torque.

The dynamics of HCDRs can be expressed by the following extended EoM from (3) and (4):

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\dot{\boldsymbol{q}}, \boldsymbol{q}) + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{a}, \tag{11}$$

$$\boldsymbol{B}(\boldsymbol{q}) = \begin{bmatrix} -\boldsymbol{L}(\boldsymbol{q})^T & \boldsymbol{A}(\boldsymbol{q}) \end{bmatrix}, \tag{12}$$

$$\underline{\boldsymbol{a}} \leq \boldsymbol{a} \leq \overline{\boldsymbol{a}} \tag{13}$$

where $\boldsymbol{B}(\boldsymbol{q}) \in \mathbb{R}^{n \times (m+p)}$ is formed by the joint-actuator Jacobian matrix $\boldsymbol{L}(\boldsymbol{q})$ and the Jacobian matrix for active joint torques $\boldsymbol{A}(\boldsymbol{q}) \in \mathbb{R}^{n \times p}$. When $p = 0$, the extended EoM of HCDRs defined by (11) reduces to the EoM for MCDRs or SCDRs in (3). Therefore, the extended EoM given by (11) will be used to describe the dynamics of CDPRs in the remainder of this article.

D. Control Objective

It is noted that most tracking tasks are defined in the operational space, but due to redundancy coming from CDPRs (referring to MCDRs and HCDRs), a good operational tracking performance does not guarantee feasible joint space performance. When the actuation space is considered to minimize the control effort, the control design becomes more challenging. In addition, undesirable situations, such as cable-link interference, loss of manipulability, and reaching mechanical joint limits, might happen. The aim of this work is thus to exploit the two-level redundancy of CDPRs to achieve good performance in the operational and joint space while minimizing the control effort and avoiding undesirable situations simultaneously. More precisely, the control objective is summarized as follows.

For a given desired trajectory $\boldsymbol{x}_d(t) \in \mathbb{R}^r$, $t \in [0, T]$ for some $T \in \mathbb{R}_{>0}$, the control objective of a CDPR is to track this desired operational space trajectory with the desired joint space performance with simultaneous minimized control effort and avoidance of undesirable situations when the task performs repetitively.

IV. Proposed Reactive-Iterative Tri-Space Operational Control Framework

This section presents a robust control framework to achieve the control objective for CDPRs. For given $\boldsymbol{x}_d(t)$, from (8), the joint space motion $\boldsymbol{q}_d(t)$ satisfies

$$\ddot{\boldsymbol{q}}_d = \boldsymbol{J}^{\dagger}(\boldsymbol{q}_d)(\ddot{\boldsymbol{x}}_d - \dot{\boldsymbol{J}}(\boldsymbol{q}_d)\dot{\boldsymbol{q}}_d) + \ddot{\boldsymbol{q}}_N. \tag{14}$$

Equation (14) composed of two parts: one is the desirable joint space trajectory $\boldsymbol{q}_d(t)$ and the other is related to the null-space trajectory $\boldsymbol{q}_N(t)$. Due to the kinematic redundancy, the solution of (14) is not unique. Optimization techniques are widely used to solve such a redundancy problem.

The cost function used in the optimization is based on the widely used lower level proportional–derivative (PD) control law, performance requirements in CDPR (MCDR and HCDR), as well as a special form of parameterization of the null-space trajectory $\boldsymbol{q}_N$.

A. PD Control Law

For a given reference trajectory in operation space $\boldsymbol{x}_d(t) \in \mathbb{R}^r$, a PD controller has been widely used. Let the tracking error be

$$\boldsymbol{e}(t) = \boldsymbol{x}_d(t) - \boldsymbol{x}(t) \tag{15}$$

where $\boldsymbol{x}$ is the actual operational space pose as defined in (5). An ideal PD controller can eliminate the nonlinearities of the CDPR, leading to the following "ideal" closed-loop system:

$$(\ddot{\boldsymbol{x}}_d - \ddot{\boldsymbol{x}}) + \boldsymbol{K}_d(\dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}}) + \boldsymbol{K}_p(\boldsymbol{x}_d - \boldsymbol{x}) = \boldsymbol{0}_r \tag{16}$$

or equivalently

$$\ddot{\boldsymbol{e}} + \boldsymbol{K}_d\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e} = \boldsymbol{0}_r \tag{17}$$

where $\boldsymbol{K}_d, \boldsymbol{K}_p \in \mathbb{R}^{r\times r}$ are diagonal matrices with positive diagonal elements. For the convenience of notation, we defined the set $\mathcal{X}_d(t) = \{\boldsymbol{x}_d(t), \dot{\boldsymbol{x}}_d(t), \ddot{\boldsymbol{x}}_d(t)\}$ containing the desired trajectory and its derivatives. Moreover, it is denoted as

$$\boldsymbol{b}(\mathcal{X}_d, \boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{e}) = \ddot{\boldsymbol{x}}_d - \dot{\boldsymbol{J}}\dot{\boldsymbol{q}} + \boldsymbol{K}_d\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e}. \tag{18}$$

Remark 1: It is noted that the choice of PD control matrices $\boldsymbol{K}_d$ and $\boldsymbol{K}_p$ will not affect the stability properties of the "ideal" closed loop, though their choices affect the transient behaviors. Even though the CDPR is working in nonideal situations, the existence of PD control law will ensure the boundedness of the operation trajectories $\boldsymbol{x}(t)$ for any $t \in [0, T]$, as shown in Theorem 2. The introduction of ILC for updating the parameters will improve the performance over iterations when the CDPR is performing a task repetitively. Hence, the choice of PD control matrices is not the focus of this work. In our analysis, these matrices are fixed. ○

Substituting $\ddot{\boldsymbol{x}} = \ddot{\boldsymbol{x}}_d + \boldsymbol{K}_d\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e}$ into (8) and writing $\ddot{\boldsymbol{q}}$ as joint acceleration command $\ddot{\boldsymbol{q}}_c(\mathcal{X}_d, \boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{e})$, (8) can be rewritten as

$$\ddot{\boldsymbol{q}}_c(\mathcal{X}_d, \boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{e}) = \boldsymbol{J}^\dagger\boldsymbol{b}(\mathcal{X}_d, \boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{e}) + \ddot{\boldsymbol{q}}_{c_N}. \tag{19}$$

It is noted that if $\ddot{\boldsymbol{q}}$ is computed, both $\dot{\boldsymbol{q}}$ and $\boldsymbol{q}$ can be obtained using numerical integration techniques. Furthermore, $\boldsymbol{e}$ is defined in operation space instead of joint space. Thus, we can rewrite $\ddot{\boldsymbol{q}}_c(\mathcal{X}_d, \boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{e})$ as $\ddot{\boldsymbol{q}}_c(\mathcal{X}_d, \ddot{\boldsymbol{q}})$ and $\boldsymbol{b}(\mathcal{X}_d, \boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{e})$ as $\boldsymbol{b}(\mathcal{X}_d, \ddot{\boldsymbol{q}})$ when optimization is designed in joint space. In an ideal case, if the operational space errors are fully compensated (i.e., $\boldsymbol{e} = \dot{\boldsymbol{e}} = 0$), (19) takes the form of (14).

Let $\boldsymbol{J}_W^\dagger = \boldsymbol{W}_q^{-1}\boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{W}_q^{-1}\boldsymbol{J}^T)^{-1}$ be the weighted pseudoinverse of $\boldsymbol{J}$ [43] with a symmetric positive definite weighting matrix $\boldsymbol{W}_q \in \mathbb{R}^{n\times n}$. The use of $\boldsymbol{W}_q$ is suitable when the joint space of the CDPR has mixed units, such as between translation and rotations. Replacing $\boldsymbol{J}^\dagger$ by $\boldsymbol{J}_W^\dagger$ in (19) yields

$$\ddot{\boldsymbol{q}}_c(\mathcal{X}_d, \ddot{\boldsymbol{q}}) = \boldsymbol{J}_W^\dagger\boldsymbol{b}(\mathcal{X}_d, \ddot{\boldsymbol{q}}) + \ddot{\boldsymbol{q}}_{c_N}. \tag{20}$$

When the joint space trajectory $\boldsymbol{q}(t)$ is designed, it is expected that $\boldsymbol{q}(t)$ can track the commanding joint acceleration $\ddot{\boldsymbol{q}}_c(\mathcal{X}_d)$ with the consideration of other joint space requirements. The error between $\ddot{\boldsymbol{q}}(t)$ and $\ddot{\boldsymbol{q}}_c(\mathcal{X}_d)$ can be formulated as one optimization problem.

B. Cost Function for Joint Space Trajectories

The user-defined cost function is denoted as $V(\mathcal{X}_d, \ddot{\boldsymbol{q}}, \boldsymbol{a})$, which is related to control objectives. In this work, it comprises three quadratic components: 1) operational space tracking function $g_t(\mathcal{X}_d, \ddot{\boldsymbol{q}})$ resolving the kinematic redundancy; 2) actuation function $g_f(\boldsymbol{a})$ minimizing control effort and resolving the actuation redundancy; and 3) the function related to constraints or avoidance function $g_a(\ddot{\boldsymbol{q}})$, to which aims undesirable situations. Hence, $V(\mathcal{X}_d, \ddot{\boldsymbol{q}}, \boldsymbol{a})$ can be expressed as

$$V(\mathcal{X}_d, \ddot{\boldsymbol{q}}, \boldsymbol{a}) = g_t(\mathcal{X}_d, \ddot{\boldsymbol{q}}) + \alpha \cdot g_f(\boldsymbol{a}) + \beta \cdot g_a(\ddot{\boldsymbol{q}}) \tag{21}$$

where $\alpha > 0$ and $\beta > 0$ are scaling factors for the actuation function and avoidance function, respectively. As a general guideline, $\alpha \ll \beta < 1$ is selected to prioritize tracking over actuation and avoidance function. Other forms of cost function can be used. The positive pair (α, β) is learned when the CDPR performs repetitive tasks.

The details of choosing the cost function $V(\mathcal{X}_d, \ddot{\boldsymbol{q}}, \boldsymbol{a})$ and other constraints in joint space will be discussed in Section V. The RC will be used to determine an appropriate joint space trajectory $\ddot{\boldsymbol{q}}(t)$ for $t \in [0, T]$.

C. Parameterization and Its Cost

Since the null-space vector $\ddot{\boldsymbol{q}}_{c_N}$ plays an important role in joint space optimization, in this work, a novel parameterization of the null-space vector $\ddot{\boldsymbol{q}}_{c_N}$ is considered such that (20) can be rewritten as

$$\ddot{\boldsymbol{q}}_{c_N} = (\boldsymbol{I} - \boldsymbol{J}_W^\dagger\boldsymbol{J})\boldsymbol{D}\boldsymbol{J}_W^\dagger\boldsymbol{b}(\mathcal{X}_d, \ddot{\boldsymbol{q}}) \tag{22}$$

where $\boldsymbol{D} = \mathrm{diag}\,(d_1, \ldots, d_n) \in \mathbb{R}^{n\times n}$ is an unknown diagonal matrix that scales the components of $\boldsymbol{J}_W^\dagger\boldsymbol{b}(\mathcal{X}_d, \ddot{\boldsymbol{q}})$, which is an n-dimensional vector. The columns of the matrix $\boldsymbol{D}$ need to be identified for improving the performance of the CDPR over repetitive tasks. The matrix $\boldsymbol{I} - \boldsymbol{J}_W^\dagger\boldsymbol{J}$ is the null-space projection matrix. Consequently, (20) is rewritten as

$$\ddot{\boldsymbol{q}}_c(\mathcal{X}_d, \ddot{\boldsymbol{q}}) = \boldsymbol{J}_W^\dagger\boldsymbol{b}(\mathcal{X}_d, \ddot{\boldsymbol{q}}) + (\boldsymbol{I} - \boldsymbol{J}_W^\dagger\boldsymbol{J})\boldsymbol{D}\boldsymbol{J}_W^\dagger\boldsymbol{b}(\mathcal{X}_d, \ddot{\boldsymbol{q}}). \tag{23}$$

Together with α and β from (21), a reactive tuning parameter vector $\boldsymbol{\theta} \in \mathbb{R}^{n+2}$ can be defined as

$$\boldsymbol{\theta} = [d_1, \ldots, d_n, \ln(\alpha), \ln(\beta)]^T. \tag{24}$$

The parameter $\boldsymbol{\theta}$ needs to be identified through repetition. Hence, the widely used technique of ILC, which improves

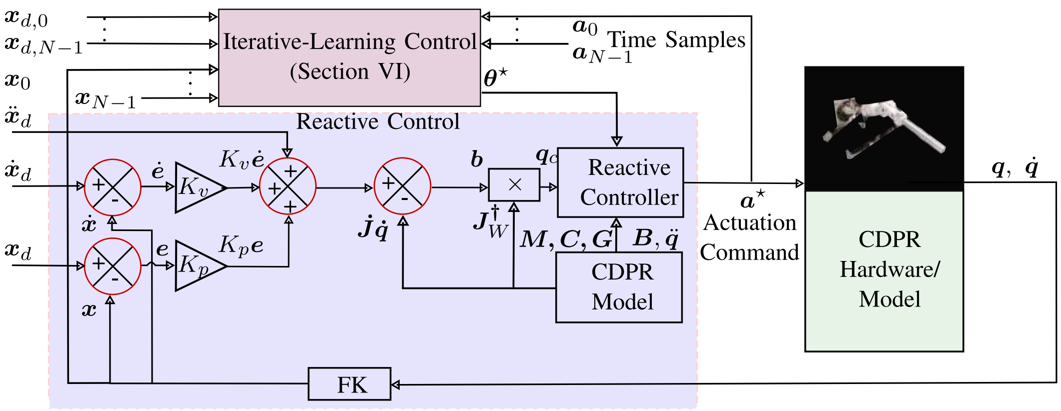


Fig. 2. Tri-space operational control framework.

the task performance through iterations when the trajectory is repeatedly executed, is used to learn an optimal θ^*.

Remark 2: It is highlighted that in this framework, two sets of optimal parameters are sought. The first set tries to balance the control effort and satisfaction of constraints, while the second optimal parameters are related to parameterization in the null space, i.e., the best parameters in the null space to achieve the optimal performance in operational space. More discussions on null-space parameterization are presented in Section VI. ○

D. Overall Structure

The proposed tri-space operational control framework consists of two primary components.

1) RC plans the joint space trajectory to solve redundancy of CDPR so that it can track the desired trajectory $x_d(t), t \in [0, T]$ in operational space and achieve the desirable performance in terms of $\ddot{q}(t)$ with respect to the cost $V(\mathcal{X}_d, \ddot{q}, a)$ and other constraints in joint space. Such a problem can be converted into an optimization problem with appropriate constraints if $\ddot{q}(t)$ and $a(t)$ are sampled with the sampling rate T_s. More precisely, at each sampling instant $t_k = 0, 1, \ldots, N-1$, an optimization problem is solved. Here, $t_k = kT_s, k = 0, 1, \ldots, N-1$, with T_s satisfying $N = (T/T_s)$. A standard QP controller is used to solve both the kinematic and actuation redundancies (tri-space redundancies) simultaneously at each sampling instant. The RC is responsible for maintaining the system stability and satisfying the unilateral constraints of $a(t)$, when encountering new trajectories without any past experiences. The reactive controller also operates real time at each time instant (faster-time scale). More details of the RC in Section V as well as Algorithm 2.
2) ILC improves the task performance through iterations by selecting the optimal parameter $\theta^\star$ with respect to some user-defined cost function $P(\cdot)$, where θ is defined in (24) when the trajectory is repeatedly executed (Section VI). To avoid confusion, the ILC cost function is referred to as the trajectory performance function (PF). The ILC focuses on improving the performance through exploring the joint space redundancy when the operational space task is repeated. The ILC performs each update at the end of each trajectory cycle (iteration) at a slower time scale.

The overall structure of tri-space operational control framework is shown in Fig. 2, where the implementation details of the RC are shown comprehensively. The ILC implementation details are discussed in Section VI. The operation modes of the control framework are given as follows.

1) Hardware Mode: In the controller operation, both hardware and simulation modes are available. In the hardware mode, the length commands are sent to the BMArm through robot operating system (ROS) at 200 Hz and cable lengths are controlled by the myomuscle units [44]. The length commands are generated from the optimized actuation command $a^\star(t_k)$, by applying a forward dynamic (FD) model. At each sampling instant t_k, joint space velocity $\dot{q}(t_k)$ and position $q(t_k)$ are determined through FK, using the cable length feedback, calculated from the motor encoder feedback (Fig. 2). This FK model is located within the CDPR/hardware block. As shown in Fig. 2, the operational space position $x(t_k)$ and velocity $\dot{x}(t_k)$ are evaluated through FK with $q(t_k)$ and $\dot{q}(t_k)$ computed using (6). To validate the cable force commands, load cells were placed in-line with the cables to measure the cable forces $f(t_k)$.

2) Simulation Mode: For a simulation mode, instead of the hardware, the length commands are sent to a CASPR CDPR model. Control frequencies of 100 and 200 Hz were selected for various simulations. During simulations, to simulate the sensor noise, additive noise was added to the model's length feedback to generate the cable lengths $l(t_k)$.

V. Reactive Controller

At each sampling instant t_k, the CDPR has joint information $q_k := q(t_k)$, $\dot{q}_k := \dot{q}(t_k)$, and $\ddot{q}_k := \ddot{q}(t_k)$. This section will first formulate the RC design as an optimization problem, followed by the discussion of the cost function $V(\mathcal{X}_d, \ddot{q}, a)$ defined in (21). Finally, Section V-B discusses the details of avoiding undesirable situations.

A. QP Optimization Formulation

The primary objective of the reactive controller is to determine a set of feasible actuation commands $a_k = a(t_k) = [f_k^T, \tau_k^T]^T$ and the reference joint trajectory $\ddot{q}_k$ that actuates the system to track the operational space trajectory $\mathcal{X}_d(t_k)$ while avoiding undesirable positions for $k = 1, 2, \ldots, N-1$ as defined by the cost function $V(\mathcal{X}_d, \ddot{\boldsymbol{q}}, \boldsymbol{a})$ with the consideration of cable force, joint torque, and EoM constraints coming from (10) to (13). This leads to the following convex QP formulation, which deals with both levels of redundancy (kinematic and actuation) in the tri-space:

$$\begin{aligned} \min_{\ddot{\boldsymbol{q}}_k, \boldsymbol{a}_k} \ & V\left(\mathcal{X}_d(t_k), \ddot{\boldsymbol{q}}_k, \boldsymbol{a}_k\right) \\ \text{s.t.} \ & \boldsymbol{M}(\boldsymbol{q}_k)\ddot{\boldsymbol{q}}_k + \boldsymbol{C}(\dot{\boldsymbol{q}}_k, \boldsymbol{q}_k) + \boldsymbol{G}(\boldsymbol{q}_k) = \boldsymbol{B}(\boldsymbol{q}_k)\boldsymbol{a}_k \\ & \underline{\boldsymbol{a}} \le \boldsymbol{a}_k \le \overline{\boldsymbol{a}} \\ & \boldsymbol{\Phi} \begin{bmatrix} \ddot{\boldsymbol{q}}_k \\ \boldsymbol{a}_k \end{bmatrix} \le \boldsymbol{\rho} \end{aligned} \tag{25}$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{n_q \times (m+n+p)}$ and $\boldsymbol{\rho} \in \mathbb{R}^{n_q}$ define the controller inequality constraints for some interger n_q.

Remark 3: The choices of $\boldsymbol{\Phi}$ and $\boldsymbol{\rho}$ are application dependent. In this work, the RC controller contains three types of linear constraints: the EoM (11), actuation bounds, and other task constraints. The EoM ensures that the system dynamics can be satisfied, while the actuation bounds ensure that the resulting actuation commands are feasible. Other inequality constraints are hard constraints that maintain the system's capability of achieving the tracking task (see Section V-B). ○

Next, we will discuss the operational space tracking $g_t(\cdot, \cdot)$ and actuation command $g_f(\cdot)$ terms of (21). The $g_a(\cdot)$ term is discussed in Section V-B2.

1) Operational Space Tracking: From (20), a good candidate for $g_t(\mathcal{X}_d, \ddot{\boldsymbol{q}})$ is

$$g_t(\mathcal{X}_d, \ddot{\boldsymbol{q}}) = \left\| \ddot{\boldsymbol{q}} - \boldsymbol{J}_W^{\dagger} \boldsymbol{b}(\mathcal{X}_d, \ddot{\boldsymbol{q}}) \right\|_2^2 \tag{26}$$

where $\boldsymbol{b}(\cdot, \cdot)$ is defined in (18). Other choices of the tracking performance can be used.

2) Actuation Command: The objective function component $g_f(\boldsymbol{a})$ is responsible for resolving the actuation redundancy for CDPR systems. As such, the actuation command $\boldsymbol{a}$ should: 1) produce the required joint acceleration $\ddot{\boldsymbol{q}}$ while satisfying the EoM (11); 2) be within the feasible actuation bound (10); and 3) satisfy some desired objectives such as minimal control effort. The EoM and actuation bounds are represented as constraints in the QP (25), and the minimization of control effort can be simply expressed by the objective function component

$$g_f(\boldsymbol{a}) = \boldsymbol{a}^T \boldsymbol{W}_a \boldsymbol{a} \tag{27}$$

where $\boldsymbol{W}_a \in \mathbb{R}^{(m+p)\times(m+p)}$ is positive-definite, typically a diagonal matrix for decoupling different joint actuation efforts in the controller.

The choice of $g_t(\cdot, \cdot)$ and $g_f(\cdot)$ was quite straightforward. In this section, from henceforth, the focus of the discussion is the avoidance of undesirable situations, which forms the foundation for the $g_a(\cdot)$ term of (21) and also defines the controller inequality constraints $\boldsymbol{\Phi}$ and $\boldsymbol{\rho}$ of (25).

B. Avoidance of Undesirable Situations

Despite the ability of the tracking task and resolution of the two-level redundancies, the system is not robust if the reactive controller does not consist of an avoidance function, to prevent the system from encountering undesirable situations, such as loss of manipulability, cable interference, and reaching mechanical joint limits.

To avoid such problems, a combination of hard and soft constraints within the reactive controller (25) is used to avoid undesirable situations. In addition to the loss of manipulability, other undesirable situations include the interference of cables and also limits on the range of joint motion. In this work, the function $h(\boldsymbol{q})$ is used to represent either constraints or the cost such as follows.

1) Satisfaction of $h(\boldsymbol{q}) \ge h_{\min}$ as a hard constraint.
2) Increase of $h(\boldsymbol{q})$ as an objective to avoid $h_{\min}$.

1) Hard Linear Constraints: In this work, the undesirable situation is represented as $h_{\min} \le h(\boldsymbol{q}) \le h_{\min} + \Delta_h$, where $\Delta_h \in \mathbb{R}^+$ represents the region size that is considered to be close to the undesirability. Since $h(\boldsymbol{q})$ is nonlinear, it is proposed to add a velocity level constraint when $h_{\min} \le h(\boldsymbol{q}) \le h_{\min} + \Delta_h$

$$\dot{h}(\boldsymbol{q}) = \frac{\partial h(\boldsymbol{q})}{\partial \boldsymbol{q}} \dot{\boldsymbol{q}} \ge \epsilon \tag{28}$$

where $\epsilon > 0$ is the minimum increase in $h(\boldsymbol{q})$ required such that the system would move away from the constraint. As the optimization variable of the QP involves $\ddot{\boldsymbol{q}}$ instead of $\dot{\boldsymbol{q}}$, the velocity can be expressed numerically as

$$\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}_p + \ddot{\boldsymbol{q}} T_s \tag{29}$$

where T_s is the time step and $\dot{\boldsymbol{q}}_p$ is the joint space velocity at the previous time step. By combining (28) with (29), the following linear constraint can be included in RC to avoid undesirable situations

$$\boldsymbol{\Phi} = \begin{bmatrix} -T_s \dfrac{\partial h}{\partial \boldsymbol{q}} & \boldsymbol{0}_{m+p} \end{bmatrix}, \quad \boldsymbol{\rho} = \frac{\partial h}{\partial \boldsymbol{q}} \dot{\boldsymbol{q}}_p - \epsilon \tag{30}$$

where $\boldsymbol{0}_{m+p}$ is an $m + p$ zero row vector. Since $\partial h(\boldsymbol{q}, \dot{\boldsymbol{q}})/\partial \boldsymbol{q}$ is a constant row vector at a particular state $\boldsymbol{q}$, so (30) is a linear constraint.

2) Avoidance Function: Hard constraints are activated such that the system can react when it is too close to undesirable situations or even failure. However, the activation of hard constraints creates sudden changes of acceleration, which potentially creates nonsmooth joint space motions. Hence, in addition to hard constraints, an avoidance function $g_a(\ddot{\boldsymbol{q}})$ is proposed to guide the system smoothly away from undesirable situations prior to reaching the hard constraints, increasing the capability to avoid undesirable situations in advance. The avoidance function $g_a(\ddot{\boldsymbol{q}})$ of (21) can be formulated as

$$g_a(\ddot{\boldsymbol{q}}) = \left\| \ddot{\boldsymbol{q}} - \ddot{\boldsymbol{q}}_A \right\|_2^2 \tag{31}$$

where $\ddot{\boldsymbol{q}}_A$ refers to the avoiding acceleration, which is the required acceleration to drive the system to avoid undesirable situations. The avoidance objective can be formulated as a desired velocity in order to increase the gradient of function $h(\boldsymbol{q})$ and the desired acceleration $\ddot{\boldsymbol{q}}_A$ expressed as

$$\ddot{\boldsymbol{q}}_A = \left(k_1 \frac{\partial h}{\partial \boldsymbol{q}}^T - \dot{\boldsymbol{q}}_p\right)\frac{1}{T_s} \tag{32}$$

where $k_1 > 0$ is a constant that governs the strength related to joint position-level avoidance.

3) Examples of Avoidance Acceleration Formulation: By introducing suitable functions for $h(\boldsymbol{q})$, the avoidance acceleration can be determined by using (32). Examples of different types of undesirable situations, their avoidance acceleration formulations, and how to combine them to determine the final avoidance acceleration are discussed as follows.

a) Loss of manipulability: The loss of manipulability $K(\boldsymbol{q})$ of a CDPR can be defined using unilateral dexterity [45]

$$K(\boldsymbol{q}) = \frac{\sigma_n(\boldsymbol{L}(\boldsymbol{q}))}{\sigma_1(\boldsymbol{L}(\boldsymbol{q}))}\frac{\sqrt{m+1}\,\eta_{\min}(\boldsymbol{q})}{\sqrt{\eta_{\min}^2(\boldsymbol{q})+1}}. \tag{33}$$

In (33), σ_1 and σ_n refer to the largest and smallest singular value of the joint-cable Jacobian matrix $\boldsymbol{L}(\boldsymbol{q})$, respectively, and $\eta_{\min}(\boldsymbol{q})$ refers to the smallest element in the normalized null-space vector $\hat{\boldsymbol{n}}(\boldsymbol{q})$, which is obtained by projecting the vector $\boldsymbol{v} = [1, 1, \ldots, 1]^T \in \mathbb{R}^m$ into the null space of $\boldsymbol{L}(\boldsymbol{q})$

$$\boldsymbol{n}(\boldsymbol{q}) = (\boldsymbol{I} - \boldsymbol{L}(\boldsymbol{q})\boldsymbol{L}^{\dagger}(\boldsymbol{q}))\boldsymbol{v} \tag{34}$$

where $\hat{\boldsymbol{n}}(\boldsymbol{q}) = \boldsymbol{n}(\boldsymbol{q})/(\|\boldsymbol{n}(\boldsymbol{q})\|)$. Consequently, $K(\boldsymbol{q})$ reaches a maximum value of 1 when the system is unilateral isotropic [45] and reaches 0 when the system loses its capability to generate any arbitrary direction of wrench through positive cable forces. Hence, the larger the $K(\boldsymbol{q})$, the higher the manipulability. Therefore, the avoidance of low manipulability acceleration $\ddot{\boldsymbol{q}}_{A_K}$ can be obtained by setting $h(\boldsymbol{q}) = K(\boldsymbol{q})$ in (32).

b) Cable-link interference: For a CDPR, the distance between each cable-link pair (SpiderArm) can be calculated numerically by treating cables and links as line segments [46], denoted as $\boldsymbol{\delta}_i(\boldsymbol{q})$. Hence, the avoidance of interference acceleration, denoted by $\ddot{\boldsymbol{q}}_{A_D}$, can be generated by setting $h(\boldsymbol{q}) = \delta_{\min}(\boldsymbol{q})$ in (32), where $\delta_{\min}(\boldsymbol{q})$ is the distance between the closest cable-link pair in the CDPR.

c) Joint limits: In MCDRs or HCDRs, hitting the mechanical joint limits is an undesirable situation, which causes instability in the system. The instability can be prevented by assuming

$$h(\boldsymbol{q}) = \begin{cases} \frac{1}{2}(\boldsymbol{q} - \boldsymbol{q}_{\max})^2, & \boldsymbol{q} \geq \boldsymbol{q}_{\max} \\ \frac{1}{2}(\boldsymbol{q} - \boldsymbol{q}_{\min})^2, & \boldsymbol{q} \leq \boldsymbol{q}_{\min} \\ 0, & \text{otherwise} \end{cases} \tag{35}$$

where $\boldsymbol{q}_{\max}$ and $\boldsymbol{q}_{\min}$ represent the maximum and minimum joint limits, respectively. The avoidance of joint limits acceleration, denoted by $\ddot{\boldsymbol{q}}_{A_L}$, can be generated by setting $h(\boldsymbol{q})$ as per (35) in (32).

d) Combining multiple undesirable situations: In cases where more than one type of undesirable situations has to be avoided, $\ddot{\boldsymbol{q}}_A$ can be defined as the weighted average of N joint accelerations

$$\ddot{\boldsymbol{q}}_A = \sum_{i=1}^{N} w_i(\boldsymbol{q}, \dot{\boldsymbol{q}}) \cdot \ddot{\boldsymbol{q}}_{a,i} \tag{36}$$

where $\ddot{\boldsymbol{q}}_{a,i}$ corresponds to the joint acceleration required to avoid the ith undesirable situations given by (32) and $w_i > 0$ represents the weights that priortize the avoidance of multiple situations. For example, if two types of undesirable situations are considered for a CDPR such as the loss of manipulability and the cable interference, then the avoiding acceleration $\ddot{\boldsymbol{q}}_A$ can be written as

$$\begin{aligned} \ddot{\boldsymbol{q}}_A &= \ddot{\boldsymbol{q}}_{A_K} + w(\boldsymbol{q})\ddot{\boldsymbol{q}}_{A_D}, \\ w(\boldsymbol{q}) &= 1/\left(1 + e^{\lambda_w(\delta_{\min}(\boldsymbol{q}) - \epsilon)}\right) \end{aligned} \tag{37}$$

where $w(\boldsymbol{q})$ is a weight function that governs the acceleration that avoids cable-link interference $\ddot{\boldsymbol{q}}_D$. In (37), the cable-link pair minimum distance $\delta_{\min}(\boldsymbol{q})$ is determined by taking the minimum of $\{\boldsymbol{\delta}_i(\boldsymbol{q})\}_{i=1}^{n_l}$, where n_l represents the number of cable-link pairs. The weight function $w(\boldsymbol{q})$ increases as the $\delta_{\min}(\boldsymbol{q})$ reaches the user-defined buffer ϵ. The rate of $w(\boldsymbol{q})$ increase is controlled by the value of λ_w. For a high value of λ_w, $w(\boldsymbol{q})$ approaches the value of unity when $\delta_{\min}(\boldsymbol{q})$ just goes below the ϵ. The weight of avoidance of low manipulability acceleration $\ddot{\boldsymbol{q}}_K$ is set to unity so that loss of manipulability avoidance can always be prioritized.

VI. Iterative-Learning Controller

ILC updates a set of control input trajectories when a system such as the CDPR is tracking a trajectory repeatedly over a finite-time interval. For CDPRs, the control input would naturally be the cable forces $\boldsymbol{f}(t) \in \mathbb{R}^m$ and joint torques $\boldsymbol{\tau}(t) \in \mathbb{R}^p$, combined to form $\boldsymbol{a}(t) \in \mathbb{R}^{m+p}$. However, due to the cable force constraints and actuation redundancy, existing ILCs cannot be directly applied onto MCDRs or HCDRs to determine the cable forces. This is partially due to the lack of a fixed or iteration invariant relationship between the control inputs and system outputs coming from redundancy [47], [48], as demonstrated in Section V. In order to solve the redundancy problem, optimization techniques have been used by various numerical gradient or search-based methods to find some optimal control input [49]. However, given the high dimensions of the problem due to the high-dimensional input signal $\mathbb{R}^{(m+p)\times N}$ for $m+p$ dimensional input and N sampling points, it is not feasible to achieve real-time control using ILC directly.

This work proposes a novel reactive-iterative tri-space operational control framework (see Section IV), which combines ILC with the reactive controller (see Fig. 2). Rather than iteratively learning the actuation command $\boldsymbol{a}(t)$ in $\mathbb{R}^{(m+p)\times N}$, it is proposed to learn a reactive tuning parameter vector $\boldsymbol{\theta} \in \mathbb{R}^{n+2}$, as discussed in Section IV. Note that the dimension of the parameter is much lower than the input vector, and hence, the "learned" parameters are used within RC.

The reactive tuning parameter vector is constant for an entire trajectory and it is designed to affect the characteristic of the controller in resolving the joint to operational space redundancy (kinematic redundancy). Hence, the joint space

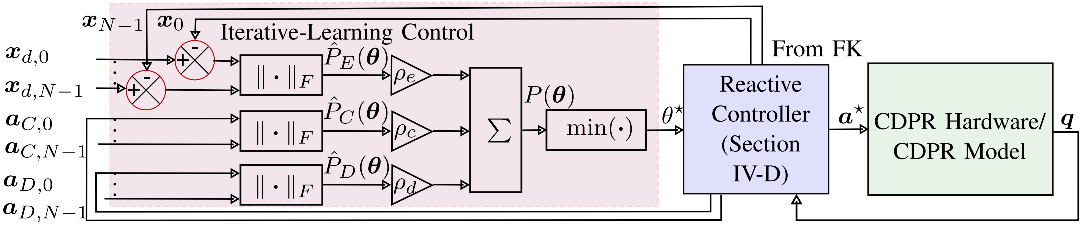


Fig. 3. ILC implementation details in the tri-space operational control framework.

profile for achieving better controller performance is explored by updating $\boldsymbol{\theta}$ while executing the desired operational space trajectory. Furthermore, the null-space exploration parameters are unconstrained and, hence, can be solved more easily. Mathematically, the ILC can be solved as an unconstrained nonlinear optimization problem

$$\boldsymbol{\theta}^\star = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^{n+2}} P(\boldsymbol{\theta}) \tag{38}$$

where $P(\cdot)$ is the trajectory PF, which will be defined later. The optimal set of parameter $\boldsymbol{\theta}^\star$ contains the optimal set of null-space parameters and the optimal α^* and β^*.

As the unconstrained optimization problem (38) is only performed once for each iteration or cycle off-line, there are many off-the-shelf optimization techniques that can be used. Practically, the parameter $\boldsymbol{\theta}$ is constrained in a known compact set $\Theta \in \mathbb{R}^{n+2}$. It is assumed that such an optimal value exists as stated in the following assumption.

Assumption 1: If the solution of the QP formulation (25) is feasible for any given initial $\boldsymbol{\theta}_0$, there exists a compact set Θ_C such that the optimal $\boldsymbol{\theta}^*$ exists uniquely in this compact set Θ_C satisfying $P(\boldsymbol{\theta}^*) = 0$.

An example of such a compact set is $\Theta_C := \{\boldsymbol{\theta} \in \mathbb{R}^{n+2} | \|\boldsymbol{\theta}\|_2 \leq C\}$, for some positive constant C. This leads to the following optimization problem:

$$\boldsymbol{\theta}^\star = \arg\min_{\boldsymbol{\theta}\in\Theta} P(\boldsymbol{\theta}). \tag{39}$$

It is usually assumed that this compact set is known.

Remark 4: It is to be noted that for a given task in the operational space and a given setting of CDPR, the optimal $\boldsymbol{\theta}^\star$ is fixed or iteration invariant. As the task is repetitive, the goal of ILC is to learn $\boldsymbol{\theta}^*$. In engineering applications, iteration-varying uncertainties and random noises exist. In this work, it is assumed that these iteration-varying uncertainties and noises are not dominant so that the task is almost repetitive. As shown in the experimental results, the proposed framework can improve the performance over iterations. ○

A. Choices of $P(\cdot)$

The evaluation of the trajectory performance, such as tracking accuracy and actuation efforts, is done after each iteration. A few performance indices (PIs) are considered. They are tracking accuracy $P_E(\cdot)$, cable actuation effort $P_C(\cdot)$, and direct actuation effort $P_D(\cdot)$ [Fig. 2(b)]. Other PIs for the trajectories over iterations can be considered.

Here, P_E represents the tracking performance, which is defined as

$$P_E(\boldsymbol{\theta}) = \|\boldsymbol{E}(\boldsymbol{\theta})\|_F \tag{40}$$

where the vector $\boldsymbol{E}$ is defined as

$$\boldsymbol{E}(\boldsymbol{\theta}) = [\boldsymbol{e}_0(\boldsymbol{\theta}), \ldots, \boldsymbol{e}_{N-1}(\boldsymbol{\theta})] \in \mathbb{R}^{r\times N}$$

where $\boldsymbol{e}$ is defined in (15). This performance $\boldsymbol{E}(\boldsymbol{\theta})$ is a function of the tracking error $\boldsymbol{e}$. The cable actuation effort $P_C(\cdot)$ is defined as

$$P_C(\boldsymbol{\theta}) = \|\boldsymbol{A_C}(\theta)\|_F \tag{41}$$

where $\boldsymbol{A_C}(\boldsymbol{\theta}) = [\boldsymbol{a}_{C,0}(\boldsymbol{\theta}), \ldots, \boldsymbol{a}_{C,N-1}(\boldsymbol{\theta})] \in \mathbb{R}^{m\times N}$ and $\{\boldsymbol{a}_{C,k}\} \in \mathbb{R}^m$ are vectors of cable actuation effort at the kth sampling instants, $\forall k = 0, \ldots, N-1$ [see (9)]. The direct actuation effort $P_D(\cdot)$ is defined as

$$P_D(\boldsymbol{\theta}) = \|\boldsymbol{A_D}(\boldsymbol{\theta})\|_F \tag{42}$$

where $\boldsymbol{A_D}(\boldsymbol{\theta}) = [\boldsymbol{a}_{D,0}(\boldsymbol{\theta}), \ldots, \boldsymbol{a}_{D,N-1}(\boldsymbol{\theta})] \in \mathbb{R}^{p\times N}$ and $\boldsymbol{a}_{D,k} \in \mathbb{R}^p$ are vectors of direct actuation effort from active joint torque at the kth sampling instants, $\forall k = 0, \ldots, N-1$ [see (9)].

Normalization: As different PFs usually vary in scale and dimension, hence they are normalized first where

$$P(\boldsymbol{\theta}) = \rho_e \hat{P}_E(\boldsymbol{\theta}) + \rho_c \hat{P}_C(\boldsymbol{\theta}) + \rho_d \hat{P}_D(\boldsymbol{\theta}) \tag{43}$$

where ρ_E, ρ_C, and ρ_D are positive normalization parameters. Here, $\hat{P}_E(\cdot)$, $\hat{P}_C(\cdot)$, and $\hat{P}_D(\cdot)$ are normalized tracking accuracy, normalized actuation effector, and normalized direct actuation effort, respectively. Usually, the normalization takes the following form, taking the example of $P_E(\cdot)$:

$$\hat{P}_E(\boldsymbol{\theta}) = \frac{P_E(\boldsymbol{\theta})}{P_E^\star}$$

where $P_E^\star = \min_{\boldsymbol{\theta}\in\Theta} P_E(\boldsymbol{\theta})$ as $P(\cdot)$ is an unknown nonlinear mapping. Here, Θ is the set containing all the computed parameters so far. Similarly, $P_C^\star = \min_{\boldsymbol{\theta}\in\Theta} P_C(\boldsymbol{\theta})$ and $P_D^\star = \min_{\boldsymbol{\theta}\in\Theta} P_D(\boldsymbol{\theta})$. These unknown parameters or values can be learned from the ILC algorithm.

In contrast to the reactive null-space projection methods that aim to optimize certain objective function at each time step [21], [50], which requires to compute the optimal $\boldsymbol{a} \in \mathbb{R}^{(m+p)\times N}$, the proposed ILC tries to update the null-space vector $\ddot{\boldsymbol{q}}_{c_N}$ [see (22)] at each iteration (slower time scale) by updating the set of null-space exploration parameters $\boldsymbol{\theta} \in \mathbb{R}^{n+2}$, defined by (24). The objective of the ILC algorithm is to identify the optimal joint space behavior, i.e., find $P_E^\star$, $P_C^\star$, $P_D^\star$, and $\boldsymbol{\theta}^\star$. The ILC implementation details are shown in Fig. 3.

B. Choice of Optimization Methods

Although the null-space exploration parameters provide a great potential in exploiting the kinematic redundancy and improving in task performance, the highly nonlinear relationship between these parameters and the trajectory PF $P(\cdot)$ is unknown, making it hard to find an optimal solution without knowing the explicit gradient information. Thus, the standard gradient-based optimization methods such as Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm [51] not applicable for solving (39) and (43).

Hence, in this work, search-based methods, such as PS [52] and PSO [53], are applied to determine $\boldsymbol{\theta}^\star$ within a given compact set Θ_C. Both techniques can find an optimal solution without the gradient information.

In simulations, PS was used as Algorithm 1 to ensure its convergence due to introduction of convergence parameter ρ_{ILC}. On the other hand, PSO is used in the experiments since PSO is well known for its ability in terms of robustness with respect to measurement noises, though the convergence of PSO is hard to show. Besides, PSO is used for the hardware mode as the hardware uncertainties (such as friction and sensor noise) affect the performance evaluation for the same set of control parameters.

Next, the PS technique is used to iteratively solve the optimization problems (39) and (43), as shown in Algorithm 1. It is known that finding $P(\boldsymbol{\theta}^\star)$ is similar to find $P_E^\star$, $P_C^\star$, and $P_D^\star$. Algorithm 1 is run three times to solve $Q_E^\star$, $Q_C^\star$, and $Q_D^\star$. Then, Algorithm 1 is run to solve the optimization problem of (43) when these values are computed. It is observed during simulations that $P_E^\star$, $P_C^\star$, and $P_D^\star$ are sensitive to the desired trajectory $\boldsymbol{x}_d$ or $\mathcal{X}_d$, and hence, they are computed once separately for a given $\mathcal{X}_d$.

Here, $\text{Proj}(\boldsymbol{\theta}, \Theta)$ constrains the updating law of $\boldsymbol{\theta}_i$ within the compact set Θ. There are many different ways to select the compact set Θ. One of them is to choose $\Theta := \{\theta_j | \underline{\theta}_j \leq \theta_j \leq \bar{\theta}_j, j = 1, \ldots, n+2\}$ so that

$$\text{Proj}\Big(\theta_j, (\underline{\theta}_j, \bar{\theta}_j)\Big) = \begin{cases} \underline{\theta}_j, & \theta \leq \underline{\theta}_j \\ \theta_j, & \underline{\theta}_j \leq \theta_j \leq \bar{\theta}_j \\ \bar{\theta}_j, & \theta_j \geq \bar{\theta}_j \end{cases}$$

which can ensure that $\boldsymbol{\theta}_i \in \Theta_C$. At each iteration, new "optimal" value $\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}^\star = \min_{s=0,\ldots,i+1}\{P(\boldsymbol{\theta}_s)\}$ is computed. Algorithm 1 will ensure that at the $(i+1)$th iteration, the PF will improve, i.e., $P(\boldsymbol{\theta}_{i+1}) \leq \rho_{\text{ILC}} P(\boldsymbol{\theta}_i)$ for some $\rho_{\text{ILC}} \in (0.1, 1)$, leading to the convergence of parameter $\boldsymbol{\theta}$ to the optimal value $\boldsymbol{\theta}^\star$. Moreover, the constraint set Θ_C used in Algorithm 1 will ensure the boundedness of the parameter updating.

Once this new optimal value is computed, it will serve as an input to the RC (Fig. 2), which uses a QP to solve the optimization problem [see (25)], as summarized in Algorithm 2. Algorithms 1 and 2 as well as Figs. 2 and 3 also show the interaction between the ILC and the RC. With the generalizabilty of the framework, other optimization methods, such as the PSO, can be easily implemented by changing the initialization procedure and the update method of $\boldsymbol{\theta}$ (Lines 1 and 5, Algorithm 1, respectively). Algorithm 2 presents the reactive controller for a single trajectory. For the convenience of notations, the set of joint space position and actuation command trajectories is denoted as $\boldsymbol{Q}$ and $\boldsymbol{A}$, respectively.

Algorithm 1 Proposed Control Framework Using PS for ILC

```
Input: Reference traj $\mathcal{X}_d = \{\boldsymbol{x}_d(t), \dot{\boldsymbol{x}}_d(t), \ddot{\boldsymbol{x}}_d(t)\} \in \mathbb{R}^r$
Input: Initial null space exploration parameters $\boldsymbol{\theta}_0 \in \mathbb{R}^{n+2}$
Input: Step size $\delta_\theta \in \mathbb{R}$
Input: The bound of $\boldsymbol{\theta}$ (the compact set $\Theta$)
Input: Convergence parameter $\rho_{ILC} \in (0, 1)$
Output: Optimal null space exploration parameters $\boldsymbol{\theta}^\star \in \mathbb{R}^{n+2}$
Output: Optimal joint space position trajectory $\boldsymbol{Q}^\star \in \mathbb{R}^{n \times N}$
Output: Optimal actuation commands traj $\boldsymbol{A}^\star \in \mathbb{R}^{(m+p) \times N}$
 1: $P^\star \in \mathbb{R} \leftarrow \infty$ (initialise $P^\star$)
 2: for $i$-th iteration of ILC do
 3:     // Explore neighborhood of $\boldsymbol{\theta}_i = \left[\theta_{i,1}, \cdots, \theta_{i,n+2}\right]^T$
 4:     for its $j^{th}$ element $\theta_{i,j}$ do
 5:         $\bar{\theta}_{i,j} \leftarrow \theta_{i,j} + \delta_\theta$ (update of $\boldsymbol{\theta}$)
 6:         // Execute trajectory with updated $\bar{\boldsymbol{\theta}}_i$
 7:         Run Algorithm 2 with $\mathcal{X}_d$, $\bar{\boldsymbol{\theta}}_i$ to obtain $\boldsymbol{Q}, \boldsymbol{A}$
 8:         $P(\bar{\boldsymbol{\theta}}_i) \leftarrow$ (43) using $\boldsymbol{Q}(\bar{\boldsymbol{\theta}}_i)$ and $\boldsymbol{A}(\bar{\boldsymbol{\theta}}_i)$
 9:     end for
10:     $P_i^\star \leftarrow$ best $P$ in the $i$-th iteration
11:     $\boldsymbol{\theta}_i^\star \leftarrow$ corresponding $\bar{\boldsymbol{\theta}}_i$ to $P_i^\star$
12:     if $P_i^\star \leq \rho_{ILC} P^\star$ and $\boldsymbol{\theta}_i \in \Theta$ then
13:         $\boldsymbol{\theta}^\star \leftarrow \boldsymbol{\theta}_i^\star$;
14:         $\boldsymbol{Q}^\star \leftarrow \boldsymbol{Q}(\boldsymbol{\theta}^\star)$;
15:         $\boldsymbol{A}^\star \leftarrow \boldsymbol{A}(\boldsymbol{\theta}^\star)$;
16:     else
17:         $\delta_\theta \leftarrow \frac{\delta_\theta}{2}$
18:         $\boldsymbol{\theta}_i \leftarrow Proj(\boldsymbol{\theta}_i, \Theta)$
19:     end if
20: end for
21: return $\boldsymbol{\theta}^\star, \boldsymbol{Q}^\star, \boldsymbol{A}^\star$
```

Algorithm 2 Reactive Controller for a Single Trajectory

```
Input: Reference traj $\mathcal{X}_d = \{\boldsymbol{x}_d(t), \dot{\boldsymbol{x}}_d(t), \ddot{\boldsymbol{x}}_d(t)\} \in \mathbb{R}^{3r}$
Input: Null space exploration parameters $\boldsymbol{\theta}_0 \in \mathbb{R}^{n+2}$
Output: Joint space position trajectory $\boldsymbol{Q}$
Output: Actuation commands trajectory $\boldsymbol{A}$
 1: for $k \leftarrow 0, \ldots, N-1$ do
 2:     $\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i \leftarrow$ from robot state
 3:     $g_t(\mathcal{X}_d, \ddot{\boldsymbol{q}}, \boldsymbol{\theta}) \leftarrow$ (26)
 4:     $g_f \leftarrow$ (27)
 5:     $g_a \leftarrow$ (31) - (36)
 6:     $\boldsymbol{\Phi}, \boldsymbol{\rho} \leftarrow$ (30)
 7:     $\ddot{\boldsymbol{q}}_i, \boldsymbol{a}_i \leftarrow$ solving (25) using $\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i, g_t, g_f, g_a, \boldsymbol{\Phi}, \boldsymbol{\rho}$
 8:     Use $\ddot{\boldsymbol{q}}_i, \boldsymbol{a}_i$ to execute on the robot
 9:     Add $\boldsymbol{q}_i$ to $\boldsymbol{Q}$
10:     Add $\boldsymbol{a}_i$ to $\boldsymbol{A}$
11: end for
12: return $\boldsymbol{Q}, \boldsymbol{A}$
```

VII. Stability Analysis

The stability of the class of controllers that minimizes the actuation effort given by (27) is a well-studied topic [54], [55] for unconstrainted cases. In particular, the gradient of the QP problem (25) without constraints is computed as the RC law $\boldsymbol{a}$ to show the stability. However, due to the existence of constraints, such an analysis is not sufficient to show the boundedness of the trajectories.

In this work, perturbation theory [56, Ch. 10] is applied for the stability analysis. The analysis shows the boundedness of trajectories for any sampling instant $k = 0, \dots, N-1$ and for any iteration, when both the RC law in (25) (see Algorithm 2) and the ILC law (see Algorithm 1) are working together. Then, the ILC algorithm (Algorithm 1) can ensure the tracking convergence. Hence, the algorithm's convergence guarantees tracking performance improvement, which can be seen in Table II.

A. Boundedness of Trajectories

At the ith iteration, any computed $\boldsymbol{\theta}_i$ from Algorithm 1 is constrained in the compact set Θ. Consequently, it leads to the bounded solutions of QP problem $\boldsymbol{a}_k^i$ and $\ddot{\boldsymbol{q}}_k^i$ [see (25) and Algorithm 2]. By applying a perturbation theory [56, Ch. 10] for a finite interval, the boundedness of $\ddot{\boldsymbol{q}}_k^i$ leads to the boundedness of $\dot{\boldsymbol{q}}_k^i$ and $\boldsymbol{q}_k^i$. Hence, at the ith iteration and within a fixed sampling time T_s, the sampled-data structure of actuation $\boldsymbol{a}^i(t)$ in (11) becomes

$$\boldsymbol{M}(\boldsymbol{q}^i(t))\ddot{\boldsymbol{q}}^i(t) + \boldsymbol{C}(\dot{\boldsymbol{q}}^i(t), \boldsymbol{q}^i(t)) + \boldsymbol{G}(\boldsymbol{q}^i(t)) = \boldsymbol{B}(\boldsymbol{q}^i(t))\boldsymbol{a}^i(t)$$
$$\text{where, } \boldsymbol{a}^i(t) = \boldsymbol{a}_k^i \quad \forall t \in [kT_s, (k+1)T_s) \qquad (44)$$

for all $k = 0, \dots, N-1$. In (44), $\boldsymbol{a}_k^i$ is computed from Algorithms 1 and 2. The continuity of the trajectories from (11) ensures that $\boldsymbol{a}^i(t)$, $\boldsymbol{q}^i(t)$, $\dot{\boldsymbol{q}}^i(t)$, and $\ddot{\boldsymbol{q}}^i(t)$ are uniformly bounded for any $t \in [0, T]$ for the ith iteration. This leads to the following theorem.

Theorem 1: Assume that a CDPR system (11) has a sampled-data structure (44) with constraints (11)–(13). If actuation $\boldsymbol{a}_k^i$ is computed from Algorithms 1 and 2, then $\boldsymbol{q}^i(t)$, $\dot{\boldsymbol{q}}^i(t)$, and $\ddot{\boldsymbol{q}}^i(t)$ in (44) are uniformly bounded for any $t \in [0, T]$ for any ith iteration. Moreover, constraints (13) are satisfied from QP (Algorithm 2). ○

Remark 5: It is noted that a PD controller is used in the designed framework. By tuning the gains of PD controller, it is possible to tune the tracking error at the ith iteration, the kth sampling instant $\boldsymbol{e}_k^i, i = 0, \dots; k = 0, \dots, N-1$, to an appropriate range. This indicates that $\boldsymbol{x}^i(t)$ is uniformly bounded for any iteration and any $t \in [0, T]$. By using pseudoinverse J, we can also conclude that $\boldsymbol{q}(t)$ is also bounded. Such an analysis as in [54] and [55] can provide less conservative estimation of the bound of trajectories. As QP (27) already provided bounded solutions, the presented analysis is simpler. It is highlighted that the performance improvement along the iteration domain is achieved by Algorithm 1. The role of RC is to avoid unwanted behaviors in time domain $t \in [0, T]$, which is achieved by QP (Algorithm 2). ○

B. Convergence of ILC Algorithm

By introducing convergence parameter ρ_{ILC} and the projection operator $\text{Proj}(\boldsymbol{\theta}, \Theta)$, the proposed PS-based ILC Algorithm 1 ensures that the cost function $P(\cdot)$ decreases monotonically. If Assumption 1 holds, Algorithm 1 can ensure that $\lim_{i\to\infty} P(\boldsymbol{\theta}_i^\star) = 0$ and $\lim_{i\to\infty} \boldsymbol{\theta}_i^\star = \boldsymbol{\theta}^\star$, indicating that $\lim_{i\to\infty} E(\boldsymbol{\theta}_i^\star) = 0$, which comes from (41) and (43). This shows that $\lim_{i\to\infty} \boldsymbol{e}_k^i(t) = \boldsymbol{0}, \forall k = 0, \dots, N-1$.

For any given $\delta > 0$, by applying [57, Th. 1], it can be shown that $\limsup_{i\to\infty} \left\|\boldsymbol{e}^i(t)\right\|_2 \le \delta, \forall t \in [0, T]$, by selecting a sufficiently small sampling period T_s. This is summarized in the following theorem.

Theorem 2: Assume that Assumption 1 holds for some known compact set Θ_c. Let δ be any positive constant. There exists $T_s^\star > 0$ such that for any small sampling interval satisfying $T_s \in (0, T_s^\star)$, the proposed Algorithm 1 can ensure that $\limsup_{i\to\infty} \left\|\boldsymbol{e}^i(t)\right\|_2 \le \delta, \forall t \in [0, T]$. ○

Remark 6: Theorem 2 shows that if Assumption 1 holds, by using Algorithm 1, the operational space tracking error $\boldsymbol{e}^i(t)$ can be arbitrarily small as the iteration number approaches to infinity. Consequently, Theorem 2 also suggested that for a given positive constant $\delta_1 > \delta$, where δ comes from Theorem 2, there exists $T_s^\star > 0$ and $N^\star \in \mathbb{N}$ such that

$$\left\|\boldsymbol{e}^i(t)\right\|_2 \le \delta \quad \forall t \in [0, T], i \in \mathbb{N} \text{ and } i \ge N^\star$$

to ensure the convergence in a finite number of iterations. ○

Remark 7: If there exists $\boldsymbol{\theta}^\star$ such that $P(\boldsymbol{\theta}^\star) \neq 0$, Algorithm 1 can select $\rho_{\text{ILC}} = 1$ to show that $\limsup_{i\to\infty} P(\boldsymbol{\theta}_i) = C_1$ for some positive constant C_1. The tracking error at steady state in iteration $\limsup_{i\to\infty} \boldsymbol{e}^i(t)$ is still bounded. Such a bound is determined by the parameters ρ_e, ρ_c, and ρ_d in (43) as well as the value of C_1. ○

VIII. Simulation Results

Simulation results on one MCDR, BMArm, and two HCDRs, SpiderArm and FASTKIT, are presented in this section to demonstrate the capability and features of the proposed framework.

In addition, the simulation results also signify the ability of the proposed reactive controller to be applied to different systems with no controller tuning required. Table I provides all the parameters associated with the simulations. Frequencies 100 and 200 Hz were selected to evaluate the controller performance with different frequencies.

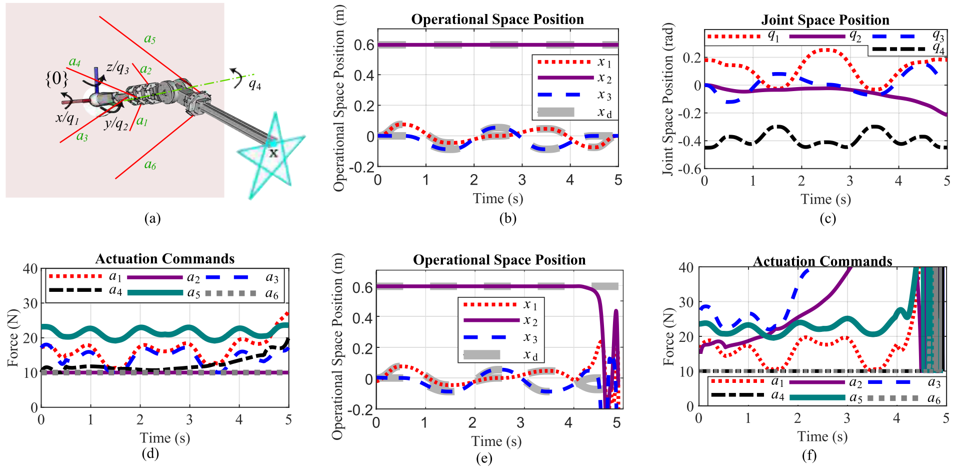
Controller gains $\boldsymbol{K}_p$ and $\boldsymbol{K}_d$ were selected in such a manner so that the damping ratio was approximately set to unity in all the cases and the boundedness of $\boldsymbol{e}(t), t \in [0, T]$ were achieved. For the ILC optimization in the simulations, PS was used as discussed in Section VI-B. All simulations were performed with the open-source CASPR [30] software using a computer with an Intel Core i9-11900K CPU @ 3.50 GHz and 32.0 GB of RAM, using MATLAB R2018a (64-bit). The QP in the reactive controller was solved using qpOASES [58].

A. BMArm

The BMArm [Fig. 4(a)] is a 4-DoF MCDR ($n = 4$). It consists of one spherical joint and one revolute joint. The

TABLE I
CONTROL AND TRAJECTORY PARAMETERS USED FOR SIMULATIONS

Row	CDPR	Fig.	α	β	$\boldsymbol{K}_p$	$\boldsymbol{K}_d$	Initial Joint Position $\boldsymbol{q}_0$	Initial Task Position $\boldsymbol{x}_0$	Frequency (Hz)	Result
1	BMArm	4 (b) to (d)	1×10^{-6}	0	$200\boldsymbol{I}_3$	$28\boldsymbol{I}_3$	$[0.18, 0, 0, -0.45]^T$	$[0, 0.595, 0]^T$	100	Success
2		4 (e) to (f), and 5 (a)	1×10^{-6}	0	$200\boldsymbol{I}_3$	$28\boldsymbol{I}_3$	$[0.18, 0.2, 0, -0.45]^T$	$[0, 0.595, 0]^T$	100	Failure
3		5 (a) to (c)	1×10^{-6}	1×10^{-2}	$200\boldsymbol{I}_3$	$28\boldsymbol{I}_3$	$[0.18, 0.2, 0, -0.45]^T$	$[0, 0.595, 0]^T$	100	Success
4	SpiderArm	6 (b) to (l)	1×10^{-6}	2×10^{-1}	Position: $40\boldsymbol{I}_3$ Orientation: $5\boldsymbol{I}_3$	Position: $12.6\boldsymbol{I}_3$ Orientation: $4.5\boldsymbol{I}_3$	$[0, 0, -\pi/2]^T$	$[2, 1.1, 1.14]^T$	100, 200	Success
5	FASTKIT	7 (b) to (f)	1×10^{-6}	1×10^{-1}	$200\boldsymbol{I}_3$	$20\boldsymbol{I}_3$	$[0, 0, -\pi/2]^T$	$[1.5, 0, 1.35]^T$	200	Success



(a) (b) (c) (d) (e) (f)

Fig. 4. BMArm operational space tracking simulation results for star-shaped trajectory, without avoidance function, i.e., $\beta = 0$. (a) Schematic of BMArm tracking a star-shaped trajectory on the xz plane. (b) Comparison of actual and desired operational space position x with initial joint position [0.18, 0, 0, −0.45]^T. (c) Joint space position with initial joint position [0.18, 0, 0, −0.45]^T. (d) Actuation command forces with initial joint position [0.18, 0, 0, −0.45]^T. (e) Failed operational space position x with initial joint position [0.18, 0, 0, −0.45]^T. (f) Failed actuation commands with initial joint position [0.18, 0, 0, −0.45]^T.

joint space $\boldsymbol{q} \in \mathbb{R}^4$ is composed of q_1, q_2, q_3, and q_4, where q_1, q_2, and q_3 are the xyz-Euler angles of the spherical joint and q_4 is the angle of the revolute joint. The joint space is actuated by six cable forces $\boldsymbol{f} \in \mathbb{R}^6$, which forms the actuation command $\boldsymbol{a} \in \mathbb{R}^6$ ($\boldsymbol{a} = \boldsymbol{f}$). The operational space $\boldsymbol{x} \in \mathbb{R}^3$ is defined as the xyz-coordinate of the second link's tip in frame {0}.

To demonstrate the capability of resolving the kinematic and actuation redundancies with the tracking function $g_t(\mathcal{X}_d, \ddot{\boldsymbol{q}})$, actuation function $g_f(\boldsymbol{a})$, and avoidance function $g_a(\dot{\boldsymbol{q}})$, simulations on the BMArm are performed. In addition, to show the relevance of the avoidance function, the simulations are performed in two parts: with and without avoidance function.

1) RC Without Avoidance Function: A star-shaped trajectory on the xz plane [Fig. 4(a)] is tracked with control and trajectory parameters defined in Row 1 of Table I with initial joint position that is set at $\boldsymbol{q} = [0.18, 0, 0, -0.45]^T$. Fig. 4(b) shows that the reactive controller tracks the reference operational space position $\boldsymbol{x}_d$ while resolving the kinematic and actuation redundancies. The resulting joint space motion $\boldsymbol{q}$ and actuation commands $\boldsymbol{a}$ are shown in Fig. 4(c) and (d), respectively.

Next, consider the simulation of the same star trajectory, but with a different initial joint position $\boldsymbol{q} = [0.18, 0.2, 0, -0.45]^T$ in which a slight twist of the system about the y-axis has been added (Row 2, Table I). The results in Fig. 4(e) and (f) show that the operational space tracking becomes unstable and eventually diverges. The manipulability metric $K(\boldsymbol{q})$, given by (33) [black dashed line, Fig. 5(a)], reaches 0, which means that the system loses its capability to generate any direction of wrench and leads to the divergence behavior of the controller.

2) RC With Avoidance Function: The avoidance acceleration was formulated by combining the avoidance of low manipulability and joint limits acceleration in the form of (37). Fig. 5 shows the effect of the avoidance of low manipulability by setting $h(\boldsymbol{q}) = K(\boldsymbol{q})$ for the same task as in Fig. 4(e) and (f), with parameters defined in Row 3 of Table I. It can be seen from Fig. 5(b) and (c) that the tracking task is completed without divergence and significant improvement in manipulability [red line, Fig. 5(a)] has been achieved as compared to Fig. 5(a), which demonstrates the ability of the avoidance function to maintain the system's manipulability to complete the tracking task.

B. SpiderArm

The SpiderArm is an HCDR that combines a 6-DoF UR3 manipulator on the end-effector of a spatial SCDR [Fig. 6(a)]. In addition, the large translational workspace of SCDR and the installation of the robot arm lead to increased dexterity. The hybrid robot is actuated by both cables and revolute joints, forming actuation command $\boldsymbol{a} \in \mathbb{R}^{14}$, where the SCDR is actuated by eight cables $[a_1, a_2, \ldots, a_8]^T$ that are attached to the base frame, and the 6-DoF robot arm is actuated by revolute joints $[a_9, a_{10}, \ldots, a_{14}]^T$. The joint space of the SpiderArm is defined as $\boldsymbol{q} = [q_1, q_2, \ldots, q_{12}]^T$, where q_1, q_2, and q_3 refer to the translation, q_4, q_5, and q_6 refer to the Euler orientation of the SCDR in frame {0}, and the joints

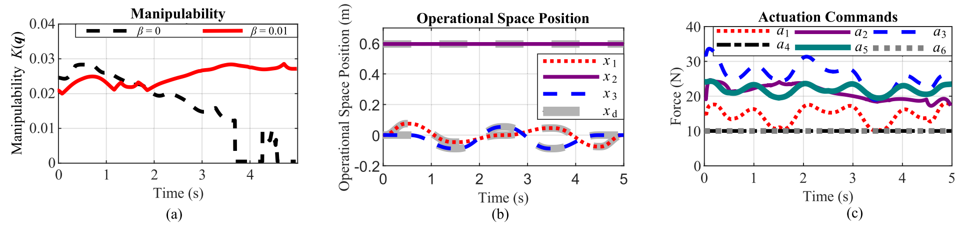


Fig. 5. BMArm operational space tracking simulation results for star-shaped trajectory, with avoidance function, i.e., $\beta = 0.01$. (a) Manipulability $K(\boldsymbol{q})$ with $\beta = 0$ and $\beta = 0.01$ with initial joint space position $[0.18, 0.2, 0.0, -0.45]^T$. (b) Successful operational space position tracking $\boldsymbol{x}$ with avoidance function, i.e., β. (c) Actuation commands a with avoidance function, i.e., $\beta = 0.01$.

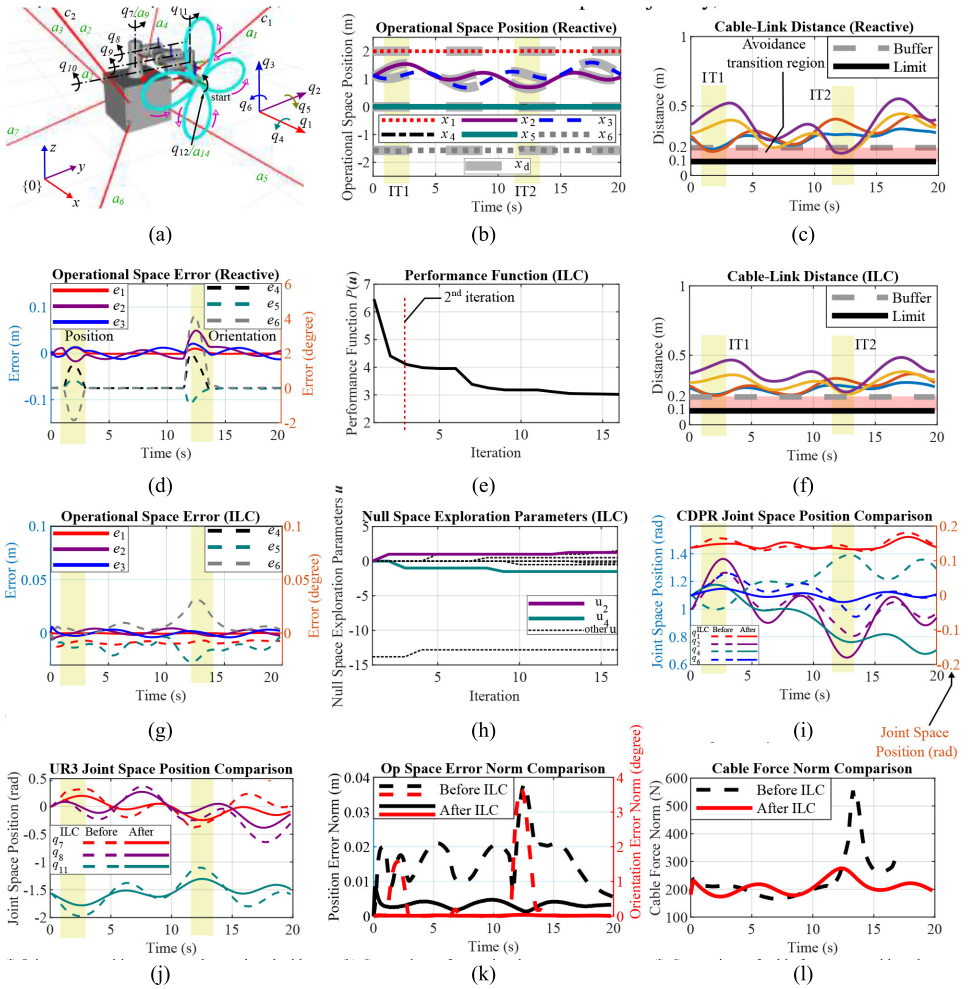


Fig. 6. SpiderArm operational space tracking simulation results for flower-shaped trajectory. (a) Schematic of the SpiderArm. (b) Operational space position and orientation $\boldsymbol{x}$ of the end-effector. (c) Cable-link distance plot for cable-link pairs, whose distance went below the buffer at IT1 and IT2. (d) Operational space position and orientation error at IT1 and IT2. (e) ILC PF reduction with iterations. (f) Cable-link interference plot with ILC tuned null-space exploration parameter u. (g) Operational space tracking error plot with ILC tuned null-space exploration parameter u. (h) Evolution of $\boldsymbol{u}$ with iterations. (i) Joint space position q_2 increased significantly with ILC. (j) Joint space position q_7, q_8, and q_{11} reduced with ILC to prevent the UR3 from hitting the cables. (k) Comparison of operational space error norm with and without ILC. (l) Comparison of cable force norm with and without ILC.

$q_7, q_8, \ldots, q_{12}$ refer to the joint angles of the robot arm. The operational space is defined as $\boldsymbol{x} = [x_1, x_2, \ldots, x_6]^T$, which represents the xyz-coordinate and Euler angles of the tip of the robot arm in frame {0}. Simulations on the SpiderArm were performed to discuss the performance of RC, ILC, and the proposed framework.

1) Reactive Control: A rose-shaped trajectory [Fig. 6(a)] was tracked [Fig. 6(b)] with parameters defined in Row 5 of Table I for $T = 20$ s to demonstrate the capability of the proposed reactive controller with avoidance acceleration.

The possibility of the SpiderArm [Fig. 6(a)] hitting the cables due to the large amplitude of the trajectory (0.45 m) influenced the consideration of two types of undesirable situations: loss of manipulability and cable-link interference (see Section V-B2), which were combined as per (37). The buffer ϵ in (37) was set at 0.2 m [see the dashed line in Fig. 6(c)], and the hard limit was set at 0.1 m [see the solid horizontal line in Fig. 6(c)], which stopped the system once encountered (see Section V-B3b). The cable-link avoidance acceleration was obtained by setting $h(\boldsymbol{q})$ to $\delta_{\min}(\boldsymbol{q})$, where $\delta_{\min}(\boldsymbol{q})$ refers to the minimum distance between the UR3 manipulator link and the closest cable.

In SpiderArm, $n_l = 48$ cable-link pairs were considered such that when one of the distance crosses below the ϵ and enters the avoidance transition region [see the red shaded region in Fig. 6(c)], the weight function $w(\boldsymbol{q})$ in (37) reaches unity abruptly, and avoidance acceleration for cable-link interference gets more weightage. For the low value of $w(\boldsymbol{q})$, the controller focused on maintaining the manipulability of the HCDR. Since the framework is generic, so other types of interference, such as cable-to-cable interference, can also be incorporated in a similar manner in the form of (37).

Fig. 6(c) shows those cable-link pairs, which crosses the buffer ϵ during the tracking task. At Interference Times IT1 and IT2 (highlighted by yellow), the distance between a cable-link pair dropped below the buffer of 0.2 m (gray dashed line), and hence, the system entered the avoidance transition region. With the avoidance function for cable interference, the controller increased the distance of the worst case cable-link pair and pushed the curve above the avoidance transition region. However, due to the weighted formulation of the QP, the need for avoidance led to reduced emphasis on tracking, leading to higher tracking error at IT1 and IT2 [Fig. 6(d)].

2) Iterative-Learning Control: To improve the tracking performance of RC, the null space of the system was explored by the proposed ILC by optimizing the trajectory PF $P(\boldsymbol{\theta})$ described in (43), which consists of the weighted sum of the normalized tracking error norm, and normalized actuation effort norm made by the cables and the joints of the CDPR and the UR3, respectively (see Section VI-A). The weights chosen for the final PF components were $\rho_E = \rho_C = \rho_D = 1$. $P(\boldsymbol{\theta})$ was optimized by the PS algorithm (Algorithm 1) with initial null-space exploration parameters $\boldsymbol{\theta}$ and step size set at $[0, \ldots 0, \ln(1 \times 10^{-6}), \ln(1 \times 10^{-2})]^T$, and unity. For $\boldsymbol{\theta} \in \mathbb{R}^{14}$, the PS algorithm has performed $14 \times 2 + 1 = 29$ evaluations in each iteration. Hence, for a total of 15 iterations, 435 (29×15) evaluations were performed.

After the optimization, $P(\theta)$ reduced significantly within the time interval of 6.5–7 s [Fig. 6(e)] since the ILC discovered a joint space trajectory that does not cause cable interference. From Fig. 6(f), the distances between cable-link pairs stayed above the buffer (gray dashed line) throughout the entire trajectory. As a result, the avoidance function was not activated during the task, and the RC emphasized more on the tracking task. Hence, the tracking error is maintained at a low level [Fig. 6(g)] and resulted in a low value of $P(\boldsymbol{\theta})$, which demonstrates that the ILC has the potential to produce a motion that both decrease operational space error, and it can avoid interference between the robot arm and the cables simultaneously, by exploiting the joint space redundancy throughout the trajectory.

In addition, further insights regarding the task were also obtained from the null-space exploration parameters. It can be observed from Fig. 6(e) that the PF experienced the most significant decrease in the first 2 iterations. From Fig. 6(h), it can be seen that the null-space exploration parameter u_2 (purple solid line) increased greatly. The increase corresponds to the importance of y translation movement of the HCDR when the kinematic redundancy is resolved. Likewise, u_4 (green solid line) associated with the rotational motion around the x-axis reduced significantly.

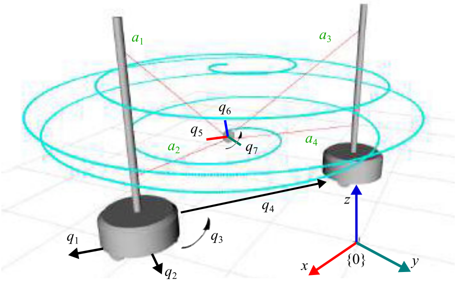


Fig. 7. FASTKIT-Planar operational space tracking simulation results for spherical-helix trajectory.

In the absence of the ILC, the UR3 manipulator link has entered the avoidance transition region at IT1 and IT2 (highlighted by yellow in Fig. 6), corresponding to the proximity of cable c_1 and c_2 [Fig. 6(a)], respectively. After the ILC implementation, the CDPR moved away from the trajectory along the x-axis since q_1 decreased at IT1 and IT2 [Fig. 6(i))]. Similarly, rotation along the x- and z-axis, given by q_4 and q_6, respectively [Fig. 6(a)] reduced significantly to prevent the manipulator from hitting the cables. The y movement of the HCDR is associated with the q_2 joint (Fig. 6(a), purple), which also increased drastically at IT1 and IT2 [Fig. 6(i)]. The increase in the translational y motion led to less required rotational movement of the UR3 q_7, q_8, and q_{11} joints [Fig. 6(j)] to cover the same trajectory workspace, thereby preventing the UR3 link from hitting the cables c_1 and c_2 by not crossing below the cable-link buffer [Fig. 6(f)].

In summary, the results show that the ILC was able to learn that for cable-link interference avoidance, the CDPR needs to move more laterally to reduce the UR3 joints' motion. Hence, the results demonstrate that changes in $\boldsymbol{\theta}$ can affect the joint space motion of the robot, and hence, it can improve task performance. In addition, the operational space error and cable force norm were also reduced after the implementation of the ILC [Fig. 6(k) and (l)).

C. FASTKIT-Planar

The FASTKIT-Planar robot [20] consists of a planar CDPR between two mobile bases [Fig. 7(a)]. The joint space is defined as $\boldsymbol{q} = [q_1, q_2, \ldots, q_7]^T$, where q_1 and q_2 refer to the xy-coordinates of the first mobile base and q_3 refers to its orientation. The distance between the two mobile bases is modeled by a prismatic joint, denoted as q_4. For the planar robot, q_5, q_6, and q_7 represent the xy-coordinates and the orientation of the end-effector in the frame of the first mobile base. The planar robot is actuated by four cables a_1, a_2, a_3, and a_4, with two cables attached on each mobile base. Actuation forces created by each joint on the mobile bases are denoted as a_5, a_6, a_7, and a_8. The operational space is defined as $\boldsymbol{x} = [x_1, x_2, x_3]^T$, which represents the xyz-coordinate of the end-effector of the planar robot in frame $\{0\}$. Next, the results obtained from the simulations performed on the FASTKIT-Planar HCDR are discussed.

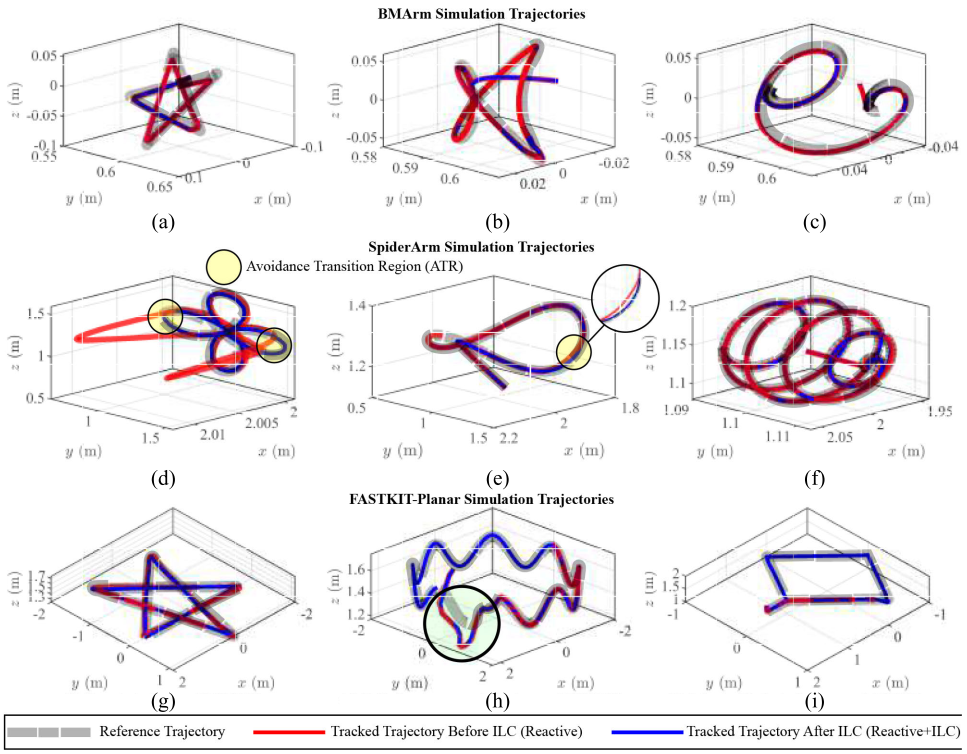


Fig. 8. Simulation results of various trajectories with added noise to verify the control framework robustness. (a) Star trajectory on BMArm. (b) Toroidal trajectory on BMArm. (c) Spherical helix trajectory on BMArm. (d) Rose trajectory on SpiderArm. (e) Cylindrical sine wave trajectory on SpiderArm. (f) Spherical helix trajectory on SpiderArm. (g) Star trajectory on FASTKIT. (h) Cylindrical sine wave trajectory on FASTKIT. (i) Rectangular trajectory on FASTKIT.

The design of the FASTKIT-Planar [Fig. 7(a)] adds two major challenges in the RC. First, the mobile bases have unlimited workspace, and their movements can possibly have no effect on the end-effector. Second, the controller should keep the movement of the planar CDPR between the two mobile bases. Hence, for a system such as the FASTKIT-Planar with a high number of redundant DoF, the QP formulation (25) for the reactive controller allows the resolution of kinematic and actuation redundancies efficiently.

In the FASTKIT-Planar, the avoidance acceleration given by (32) was formulated by combining the avoidance of low manipulability and cable-link interference acceleration as per (37). To prevent the planar robot from colliding with the mobile bases, the avoidance acceleration for cable-link interference was obtained by setting $h(\boldsymbol{q})$ to $\delta_{\min}(\boldsymbol{q})$, where $\delta_{\min}(\boldsymbol{q})$ refers to the miniimum distance between the planar end-effector and the closest mobile base. The kinematic redundancy on the FASTKIT-Planar robot was explored and exploited with the use of the ILC on various trajectories. The FASTKIT-Planar simulations are discussed in Section VIII-D.

D. Robustness of the Proposed Framework With Various Trajectories

In Sections VIII-A–VIII-C, detailed results were presented to demonstrate the behavior of the proposed framework. However, many more simulations were also conducted by adding white Gaussian noise to show the robustness of the proposed framework. Hence, in this section, a dataset of many different trajectories were produced, and before and after ILC implementation, simulations were run on them in the presence of added noise to simulate the real-time sensor noise. For simulation comparison, the selected PIs are operational space root-mean-square error (RMSE) in tracking in x-, y-, and z-directions, average cable force norm, and average direct actuation norm. All the simulation parameters were set as per the values set in Section VIII-A, VIII-B, and VIII-C. The performance results are compared in Fig. 8 and tabulated in Table II. Next, we summarize the simulation results for the three CDPRs.

1) BMArm: Table II shows the RMSE and average cable force norm values for the BMArm. It showed that these values decreased slightly for all trajectories [Fig. 8(a)–(c)]. For the toroidal trajectory [Fig. 8(b)], the manipulability increased significantly from 3.37×10^{-2} to 4.10×10^{-1}.

2) SpiderArm: From Table II, it was observed that the RMSE for the SpiderArm tracking either decreased (Spherical helix) or remained almost the same (rose and cylindrical sine wave) before and after the ILC implementation. However, the ILC was able to improve the tracking error significantly at the IT (encircle and highlighted by yellow in Fig. 8(d) and (e)) by discovering a joint space trajectory, which would not allow the system to enter into the avoidance transition region. In addition, the average cable force and direct actuation norms also decreased greatly.

3) FASTKIT-Planar: In the FASTKIT-Planar, after the ILC implementation, all the PIs decreased significantly [Fig. 8(g)] for the star-shaped trajectory. However, with noise and without the ILC, the simulations shown in Fig. 8(h) and (i) failed for the FASTKIT-Planar, which became successful with the ILC. In addition, Fig. 8(h) shows that in the presence of random

TABLE II

COMPARISON OF BEFORE AND AFTER ILC IMPLEMENTATION IMPACT ON THE PIS WITH NOISE

CDPR	Trajectory with noise		Root mean square tracking error (m)						Average cable force norm (N)		Average direct actuation norm (N)		Average computation time (s)	
	Name	Fig. 8	Before ILC			After ILC			Before ILC	After ILC	Before ILC	After ILC	Controller	Net
			x	y	z	x	y	z						
BMArm	Star	(a)	9.87e-03	4.58e-04	1.16e-02	9.64e-03	4.49e-0	1.16e-02	45.6	45.0	-	-	0.012	1.38
	Toroidal	(b)	1.16e-03	1.85e-04	1.20e-03	7.77e-04	1.68e-04	1.20e-03	45.6	44.8	-	-	0.012	3.53
	Spherical helix	(c)	4.30e-03	3.20e-04	5.12e-03	4.58e-03	3.15e-04	5.08e-03	43.5	43.4	-	-	0.012	3.37
SpiderArm	Rose	(d)	3.57e-03	1.41e-02	9.03e-03	5.05e-04	1.92e-03	1.96e-03	263	208	18.4	18.8	0.014	5.83
	Cylinder sine wave	(e)	1.90e-03	7.25e-03	2.05e-03	1.94e-03	7.73e-03	2.08e-03	265	192	18.6	15.4	0.006	6.64
	Spherical helix	(f)	5.26e-03	5.41e-03	5.96e-03	0.80e-03	5.32e-03	0.93e-03	183	152	19.2	19.4	0.006	2.29
FASTKIT	Star	(g)	3.40e-02	2.94e-02	4.59e-02	2.85e-02	1.36e-02	4.59e-02	23.5	21.6	126	91	0.006	2.29
	Cylinder sine wave	(h)	Fail	Fail	Fail	6.75e-02	4.44e-02	5.17e-02	265	37.7	Fail	43.8	0.007	6.64
	Rectangular	(i)	Fail	Fail	Fail	6.68e-03	2.80e-03	3.99e-02	Fail	46.1	Fail	54.5	0.006	2.29

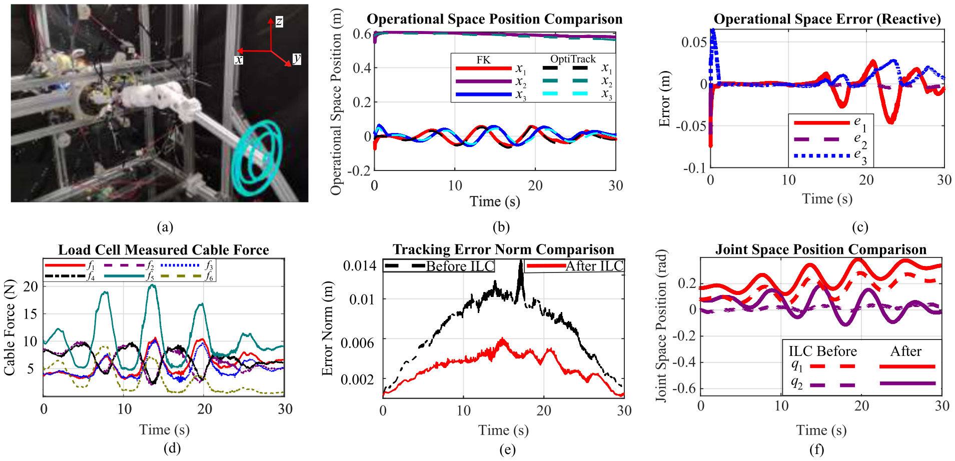


Fig. 9. BMArm hardware operational space tracking results for spherical-helix trajectory. (a) BMArm robot tracking a spherical-helix trajectory. (b) Tracking comparison between the FK-based, and Opti-Track captured position with the reactive controller. (c) Tracking error comparison between the BMArm and the reference without ILC. (d) Cable forces measured by the load cells during the tracking task. (e) Comparison of the tracking error norm with and without ILC. (f) Comparison of the joint space position with and without ILC.

noise, the tracking initially started poorly (encircled), but after $t = 1.1$ s, the system's end-effector movement resumed the reference trajectory path. For Fig. 8(h) and (i), the reactive controller tracking failed at $t = 10$ and $t = 5$ s. However, the ILC operated reactive controller was able to finish the tracking without divergence.

The analysis of the results shows that the selection of α and β is important for feasibility and performance of the controller framework. By referring to (21), a low value of α and β emphasized more on minimizing tracking error. For $\alpha = \beta = 0$, a significant reduction in the RMSE was found. However, for simulation with $\alpha = 100$, a greater reduction in the average cable force and direct actuation norms were observed.

IX. HARDWARE RESULTS

The proposed control framework was applied on the BMArm robot hardware [Fig. 9(a)]. The hardware was controlled by two computers: 1) a 64-bit computer, with a 3.4-GHz Core i7-6700 processor and 16-GB RAM, running the CASPR-ROS software platform [59] responsible for the low-level motor control, and 2) with an Intel Core i9-11900K CPU @ 3.50 GHz and 32.0 GB of RAM, running CASPR in MATLAB for the proposed reactive and ILC framework.

A. Reactive Control

A helix-shaped trajectory, similar to that in Section VIII-C, was tracked, rotated about the y-axis with a radius of 0.06 m, a thickness of 0.012 m, and center at $[0, 0.595, 0]^T$. The starting joint pose was set at $\boldsymbol{q} = [0.08, 0, 0, -0.2]^T$. Control parameters were set at $\boldsymbol{K}_p = 20\,000\boldsymbol{I}_3$, $\boldsymbol{K}_d = 282\boldsymbol{I}_3$ (damping ratio ≈ 1), $\alpha = 1 \times 10^{-6}$, and $\beta = 1$. A high value was chosen for $\boldsymbol{K}_p$ to overcome the friction in the system caused by the routing of cables through the various pulleys.

The BMArm's actual operational space position $\boldsymbol{x}$ was obtained through FK using the cable length feedback, which was calculated from the motor encoder feedback. To validate the FK determined position, Fig. 9(b) compares the FK-based end-effector position with the position captured by the OptiTrack Prime 13 system. Fig. 9(c) shows the operational space tracking error between the hardware tracking task and the reference trajectory in which an error amplitude lower than 0.05 m was obtained. Cable forces measured by the load cells [Fig. 9(d)] also show that the tracking task was achieved with positive cable forces. For the computational time of the reactive controller, a worst case of 15.5 ms was recorded during the warm-up stage of the QP process. However, 99% of the time steps achieved an average of 2.2 ms, which demonstrates the capability of the proposed reactive controller to achieve tri-space operational control of CDPR hardware online in real time. The obtained results are consistent with the BMArm simulations.

B. Iterative-Learning Control

ILC was also performed on the BMArm hardware as per Section VI. For optimizing (39) and (43), PSO with six particles was used to explore the solution space of $\boldsymbol{\theta}$ over 15 iterations, resulting $6 \times 15 = 90$ evaluations. Position bounds for the particles were set at $\boldsymbol{\theta}_{\min} = [-2, -2, -2, -2, \ln(1 \times 10^{-7}), \ln(0.1)]^T$, $\boldsymbol{\theta}_{\max} = [2, 2, 2, 2, \ln(1 \times 10^{-5}), \ln(10)]^T$. The inertia weight ω was set at 0.73, while particle and swarm best parameters ϕ_p and ϕ_g were both set at 1.5 [53].

The results with and without ILC show that the tracking has improved, with the maximum amplitude of the error norm reducing from 0.014 to 0.006 m [Fig. 9(e)]. In addition, the cable forces used in the tracking task also reduced along the entire trajectory. Both improvements in the tracking error and cable force norm contributed a 64% reduction in the PF.

The kinematic redundancy was resolved by the RC, which resulted in the joint space trajectory shown by the dashed lines in Fig. 9(a). Furthermore, by comparing the joint space trajectory before and after ILC implementation, it can be observed that the tracking error and cable force norm improvement are closely related to an increase in the movement of q_2, which corresponds to the twisting motion along the y-axis. Although excessive twisting creates a potential risk for loss of manipulability (Section V-B), the ILC could determine a joint space trajectory that allowed twisting to a suitable extent such that the tracking error and cable force needed for the task were reduced while remaining stable. The average computational time recorded for updating $\boldsymbol{\theta}$ and evaluating $P(\boldsymbol{\theta})$ after completing each trajectory is 0.75 and 16.71 ms, respectively, which implies that the proposed control framework could complete a new task and improve the task performance online on robot hardware.

X. Conclusion

This work proposed a novel tri-space operational control framework, dealing with actuation and kinematic redundancies, was proposed for a class of CDPR, including MCDR and HCDR when they are performing repetitive tasks. This framework consisted of both an RC and an ILC systematically. The RC was used for solving a convex QP problem such that an operational space reference trajectory can be tracked with feasible cable forces while avoiding undesirable situations. The ILC explored the best set of null-space exploration parameters in the null-space component of the joint space. It also tuned the objective function weights. Hence, the ILC component further improved the performance of the RC at a trajectory level. To demonstrate the capability and generalizability of the proposed framework, simulations were performed on one MCDR, BMArm, and two HCDRs, SpiderArm and FASTKIT-Planar robot. Experimental results on a two-link MCDR of BMArm were also presented to demonstrate the practicality of the framework in controlling hardware online.

References

[1] J. Albus, R. Bostelman, and N. Dagalakis, "The NIST robocrane," *J. Feild Robot.*, vol. 10, no. 5, pp. 709–724, 1993.

[2] S.-R. Oh, K. Mankala, S. K. Agrawal, and J. S. Albus, "A dual-stage planar cable robot: Dynamic modeling and design of a robust controller with positive inputs," *J. Mech. Des.*, vol. 127, no. 4, pp. 612–620, 2005.

[3] P. Bosscher, R. L. Williams, L. S. Bryson, and D. Castro-Lacouture, "Cable-suspended robotic contour crafting system," *Autom. Construct.*, vol. 17, no. 1, pp. 45–55, Nov. 2007.

[4] Y. Wu, H. H. Cheng, A. Fingrut, K. Crolla, Y. Yam, and D. Lau, "CU-brick cable-driven robot for automated construction of complex brick structures: From simulation to hardware realisation," in *Proc. IEEE Int. Conf. Simulation, Modeling, Program. Auto. Robots (SIMPAR)*, May 2018, pp. 166–173.

[5] Y. Mao and S. K. Agrawal, "Design of a cable-driven arm exoskeleton (CAREX) for neural rehabilitation," *IEEE Trans. Robot.*, vol. 28, no. 4, pp. 922–931, Aug. 2012.

[6] J.-B. Izard et al., "Large-scale 3D printing with cable-driven parallel robots," *Construct. Robot.*, vol. 1, no. 1, pp. 69–76, 2017.

[7] D. Lau, K. Bhalerao, D. Oetomo, and S. K. Halgamuge, "On the task specific evaluation and optimisation of cable-driven manipulators," in *Advances in Reconfigurable Mechanisms and Robots I*, J. S. Dai, M. Zoppi, and X. Kong, Eds. London, U.K.: Springer 2012, ch. 63, pp. 707–716.

[8] S. Bouchard, C. Gosselin, and B. Moore, "On the ability of a cable-driven robot to generate a prescribed set of wrenches," *J. Mech. Robot.*, vol. 2, no. 1, Feb. 2010, Art. no. 011010.

[9] M. Hassan and A. Khajepour, "Analysis of bounded cable tensions in cable-actuated parallel manipulators," *IEEE Trans. Robot.*, vol. 27, no. 5, pp. 891–900, Oct. 2011.

[10] D. Lau and D. Oetomo, "Conditions on the cable-routing matrix for wrench closure of multilink cable-driven manipulators," *J. Mech. Des.*, vol. 138, no. 3, Mar. 2016, Art. no. 032303.

[11] M. Gouttefarde, D. Daney, and J.-P. Merlet, "Interval-analysis-based determination of the wrench-feasible workspace of parallel cable-driven robots," *IEEE Trans. Robot.*, vol. 27, no. 1, pp. 1–13, Feb. 2011.

[12] S. Rezazadeh and S. Behzadipour, "Workspace analysis of multibody cable-driven mechanisms," *J. Mech. Robot.*, vol. 3, no. 2, May 2011, Art. no. 021005.

[13] A. B. Alp and S. K. Agrawal, "Cable suspended robots: Design, planning and control," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2002, pp. 4275–4280.

[14] S. Fang, D. Franitza, M. Torlo, F. Bekes, and M. Hiller, "Motion control of a tendon-based parallel manipulator using optimal tension distribution," *IEEE/ASME Trans. Mechatronics*, vol. 9, no. 3, pp. 561–568, Sep. 2004.

[15] S.-R. Oh and S. K. Agrawal, "Cable suspended planar robots with redundant cables: Controllers with positive tensions," *IEEE Trans. Robot.*, vol. 21, no. 3, pp. 457–465, Jun. 2005.

[16] M. Hassan and A. Khajepour, "Optimization of actuator forces in cable-based parallel manipulators using convex analysis," *IEEE Trans. Robot.*, vol. 24, no. 3, pp. 736–740, Jun. 2008.

[17] G. Yang, W. Lin, M. S. Kurbanhusen, C. Bang Pham, and S. Huat Yeo, "Kinematic design of a 7-DOF cable-driven humanoid arm: A solution-in-nature approach," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, Jul. 2005, pp. 444–449.

[18] S. K. Mustafa and S. K. Agrawal, "Reciprocal screw-based force-closure of an n-DOF open chain: Minimum number of cables required to fully constrain it," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 3029–3034.

[19] P. Racioppo and P. Ben-Tzvi, "Modeling and control of a cable driven modular snake robot," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, Aug. 2017, pp. 468–473.

[20] T. Rasheed, P. Long, D. Marquez-Gamez, and S. Caro, "Tension distribution algorithm for planar mobile cable-driven parallel robots," in *Cable-Driven Parallel Robots* (Mechanisms and Machine Science), vol. 53, C. Gosselin, P. Cardou, T. Bruckmann, and A. Pott, Eds. Cham, Switzerland: Springer, 2018, pp. 268–279.

[21] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE J. Robot. Autom.*, vol. RA-3, no. 1, pp. 43–53, Feb. 1987.

[22] V. De Sapio, J. Warren, O. Khatib, and S. Delp, "Simulating the task-level control of human motion: A methodology and framework for implementation," *Vis. Comput.*, vol. 21, no. 5, pp. 289–302, 2005.

[23] L. Jin et al., "Perturbed manipulability optimization in a distributed network of redundant robots," *IEEE Trans. Ind. Electron.*, vol. 68, no. 8, pp. 7209–7220, Aug. 2021.

[24] M. Shimizu, H. Kakuya, W. K. Yoon, K. Kitagaki, and K. Kosuge, "Analytical inverse kinematic computation for 7-DOF redundant manipulators with joint limits and its application to redundancy resolution," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1131–1142, Oct. 2008.

[25] M. Cheong Lei and D. Oetomo, "Modelling of cable wrapping phenomenon towards improved cable-driven mechanisms," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, Jul. 2013, pp. 649–655.

[26] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *J. Robot. Syst.*, vol. 1, no. 2, pp. 123–140, 1984.

[27] J.-X. Xu and Y. Tan, *Linear and Nonlinear Iterative Learning Control* (Lecture Notes in Control and Information Sciences), vol. 291. Berlin, Germany: Springer-Verlag, 2003.

[28] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Syst. Mag.*, vol. 26, no. 3, pp. 96–114, Jun. 2006.

[29] R. H. Chi, Z. S. Hou, B. Huang, and S. T. Jin, "A unified data-driven design framework of optimality-based generalized iterative learning control," *Comput. Chem. Eng.*, vol. 77, pp. 10–23, Jun. 2015.

[30] D. Lau, J. Eden, Y. Tan, and D. Oetomo, "CASPR: A comprehensive cable-robot analysis and simulation platform for the research of cable-driven parallel robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 3004–3011.

[31] J. Lamaury, M. Gouttefarde, A. Chemori, and P.-E. Herve, "Dual-space adaptive control of redundantly actuated cable-driven parallel robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 4879–4886.

[32] R. Qi, A. Khajepour, and W. W. Melek, "Redundancy resolution and disturbance rejection via torque optimization in hybrid cable-driven robots," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 7, pp. 4069–4079, Jul. 2022.

[33] Z. Zhang, Y. Lin, S. Li, Y. Li, Z. Yu, and Y. Luo, "Tricriteria optimization-coordination motion of dual-redundant-robot manipulators for complex path planning," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 4, pp. 1345–1357, Jul. 2018.

[34] Z. Zhang et al., "A varying-parameter convergent-differential neural network for solving joint-angular-drift problems of redundant robot manipulators," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 2, pp. 679–689, Apr. 2018.

[35] Z. Zhang and Z. Yan, "A varying parameter recurrent neural network for solving nonrepetitive motion problems of redundant robot manipulators," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 6, pp. 2680–2687, Nov. 2019.

[36] Z. Zhang, S. Chen, and S. Li, "Compatible convex–nonconvex constrained QP-based dual neural networks for motion planning of redundant robot manipulators," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 3, pp. 1250–1258, May 2019.

[37] Z. Zhang, L. Zheng, Z. Chen, L. Kong, and H. R. Karimi, "Mutual-collision-avoidance scheme synthesized by neural networks for dual redundant robot manipulators executing cooperative tasks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 3, pp. 1052–1066, Mar. 2021.

[38] D. Lau, D. Oetomo, and S. K. Halgamuge, "Inverse dynamics of multilink cable-driven manipulators with the consideration of joint interaction forces and moments," *IEEE Trans. Robot.*, vol. 31, no. 2, pp. 479–488, Apr. 2015.

[39] R. Qi, M. Rushton, A. Khajepour, and W. W. Melek, "Decoupled modeling and model predictive control of a hybrid cable-driven robot (HCDR)," *Robot. Auto. Syst.*, vol. 118, pp. 1–12, Aug. 2019.

[40] M. Michelin, P. E. Hervé, O. Tempier, J. B. Izard, and M. Gouttefarde, "Path following demonstration of a hybrid cable-driven parallel robot," in *Proc. Int. Conf. Cable-Driven Parallel Robots*. Cham, Switzerland: Springer, 2021, pp. 323–335.

[41] J. H. de Groot and R. Brand, "A three-dimensional regression model of the shoulder rhythm," *Clin. Biomechanics*, vol. 16, no. 9, pp. 735–743, Nov. 2001.

[42] D. Lau, D. Oetomo, and S. K. Halgamuge, "Generalized modeling of multilink cable-driven manipulators with arbitrary routing using the cable-routing matrix," *IEEE Trans. Robot.*, vol. 29, no. 5, pp. 1102–1113, Oct. 2013.

[43] D. E. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Trans. Man-Mach. Syst.*, vol. MMS-10, no. 2, pp. 47–53, Jun. 1969.

[44] H. G. Marques et al., "MYOROBOTICS: A modular toolkit for legged locomotion research using musculoskeletal designs," in *Proc. 6th Int. Symp. Adapt. Motion Animals Mach. (AMAM)*, 2013, pp. 1–2.

[45] R. Kurtz and V. Hayward, "Dexterity measures with unilateral actuation constraints: The n+1 case," *Adv. Robot.*, vol. 9, no. 5, pp. 561–577, Jan. 1994.

[46] V. J. Lumelsky, "On fast computation of distance between line segments," *Inf. Process. Lett.*, vol. 21, no. 2, pp. 55–61, Aug. 1985.

[47] A. Tayebi, "Adaptive iterative learning control for robot manipulators," *Automatica*, vol. 40, no. 7, pp. 1195–1203, Jul. 2004.

[48] P. Janssens, G. Pipeleers, and J. Swevers, "A data-driven constrained norm-optimal iterative learning control framework for LTI systems," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 2, pp. 546–551, Mar. 2013.

[49] T.-Y. Kuc, K. Nam, and J. S. Lee, "An iterative learning control of robot manipulators," *IEEE Trans. Robot. Autom.*, vol. 7, no. 6, pp. 835–842, Dec. 1991.

[50] V. De Sapio, O. Khatib, and S. Delp, "Least action principles and their application to constrained and task-level problems in robotics and biomechanics," *Multibody Syst. Dyn.*, vol. 19, no. 3, pp. 303–322, Apr. 2008.

[51] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Math. Program.*, vol. 45, no. 1, pp. 503–528, 1989.

[52] R. Hooke and T. A. Jeeves, "'Direct search' solution of numerical and statistical problems," *J. ACM*, vol. 8, no. 2, pp. 212–229, 1961.

[53] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Mach. Hum. Sci. (MHS)*, 1995, pp. 39–43.

[54] J. Peters and S. Schaal, "Learning to control in operational space," *Int. J. Robot. Res.*, vol. 27, no. 2, pp. 197–212, 2008.

[55] P. Hsu, J. Mauser, and S. Sastry, "Dynamic control of redundant manipulators," *J. Robotic Syst.*, vol. 6, no. 2, pp. 133–148, 1989.

[56] H. Khalil, "Perturbation theory and averaging," in *Nonlinear Systems*, 3rd ed. London, U.K.: Pearson, 2001, ch. 10, pp. 381–402.

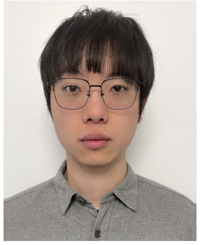
[57] D. Nešić, A. R. Teel, and P. V. Kokotović, "Sufficient conditions for stabilization of sampled-data nonlinear systems via discrete-time approximations," *Syst. Control Lett.*, vol. 38, pp. 259–270, Dec. 1999.

[58] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpOASES: A parametric active-set algorithm for quadratic programming," *Math. Program. Comput.*, vol. 6, no. 4, pp. 327–363, 2014.

[59] J. Eden, C. Song, Y. Tan, D. Oetomo, and D. Lau, "CASPR-ROS: A generalised cable robot software in ROS for hardware," in *Cable-Driven Parallel Robots* (Mechanisms and Machine Science), vol. 53, C. Gosselin, P. Cardou, T. Bruckmann, and A. Pott, Eds. Cham, Switzerland: Springer, 2018, pp. 50–61.

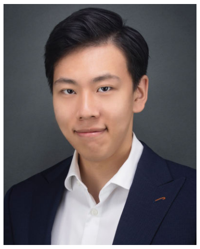
Dipankar Bhattacharya (Member, IEEE) received the B.Tech. degree in electronics and communication engineering from the North Eastern Regional Institute of Science and Technology (NERIST), Nirjuli, India, in 2010, the M.Tech. degree in electrical engineering from IIT Roorkee, Roorkee, India, in 2013, and the Ph.D. degree in mechatronics engineering from the University of Auckland, Auckland, New Zealand, in 2021.

He is currently working as a Post-Doctoral Research Fellow with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong. His research interests include biologically inspired soft robots, cable-driven parallel robots, and robotics modeling and control.



Yin Pok Chan received the B.Eng. and M.Phil. degrees in mechanical and automation engineering from The Chinese University of Hong Kong, Hong Kong, in 2017 and 2019, respectively.

His research interests include cable robots, control systems, optimization, and machine learning.



Siqi Shang received the bachelor's degree in computer science from The Chinese University of Hong Kong, Hong Kong, in 2021, and the master's degree in computer science from Columbia University, New York, NY, USA, in 2023.

His research interests include cable-driven robots, robotic learning, and dexterous manipulation.

Yuen Shan Chan received the B.Eng. and M.Phil. degrees in mechanical and automation engineering from The Chinese University of Hong Kong, Hong Kong, in 2017 and 2019, respectively. For his M.Phil. studies, he specialized in the application of cable-driven robotics, particularly in large-scale maneuvering applications.

In addition, he has worked on various projects, including underwater Remotely Operated Vehicles (ROV) and architecture perspective.

Ying Tan (Fellow, IEEE) received the bachelor's degree from Tianjin University, Tianjin, China, in 1995, and the Ph.D. degree from the National University of Singapore, Singapore, in 2002.

She is currently a Professor with the Department of Mechanical Engineering, The University of Melbourne, Parkville, VIC, Australia. In 2002, she joined the Department of Chemical Engineering, McMaster University, Hamilton, ON, Canada, as a Post-Doctoral Fellow. Since 2004, she has been with The University of Melbourne. Her research interests are in intelligent systems, nonlinear systems, real-time optimization, sampled-data systems, rehabilitation robotic systems, human motor learning, wearable sensors, and model-guided machine learning.

Dr. Tan is a fellow of the Institution of Engineers of Australia (IEAUST) and the Asia–Pacific Artificial Intelligence Association. She was awarded an Australian Post-Doctoral Fellow for the term of 2006 to 2008 and a Future Fellow for the term of 2009 to 2013 by the Australian Research Council.

Darwin Lau (Senior Member, IEEE) received the B.Eng. degree (Hons.) in mechatronics engineering and the B.CS. and Ph.D. degrees in mechanical engineering and robotics from The University of Melbourne, Parkville, VIC, Australia, in 2008 and 2014, respectively.

He was a Post-Doctoral Research Fellow with Pierre and Marie Curie University, Paris, France, from 2014 to 2015. He is currently an Associate Professor with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong. His research interests include kinematics, dynamics and control of redundantly actuated mechanisms, cable-driven parallel manipulators, construction and architectural robotics, and bioinspired robots.