Control of Operational Space with Redundancy for Multilink and Hybrid Cable-Driven Robots using a Reactive and Iterative-Learning Approach

Yin Pok Chan[1], Yuen Shan Chan[1], Darwin Lau[1], *Member, IEEE,*

***Abstract*—The operational space control of cable-driven parallel robots (*CDPRs*) is required to execute tasks in practice. This problem is challenging for CDPRs due to the two levels of redundancy and numerous constraints that exist within the actuation, joint and operational spaces. Operational space control is particularly significant for multilink cable-driven robots (*MCDRs*) and hybrid cable-driven robots (*HCDRs*) due to the fact that the dimension of the operational space is less than the number of degrees-of-freedom of the robot. MCDRs and HCDRs are developed for its higher orientation capability, dexterity and larger workspace. To the best of the authors' knowledge, there does not exist any general operational space control framework that is robust and effective for MCDRs and HCDRs. In this paper, a CDPR operational space control framework that combines reactive and iterative-learning control (*ILC*) is proposed. The framework allows the tracking of operational space trajectories online with feasible cable forces, while avoiding undesirable situations such as cable interference and loss of manipulability. It is also shown that the performance can be improved through ILC when the trajectory is repeatedly executed. Simulation and experimental hardware results on various MCDRs and HCDRs show that the proposed control framework can be conveniently and effectively applied to different types of CDPRs to be used in real-time.**

***Index Terms*—Cable-driven robots, operational space control, reactive control, iterative learning control**

I. Introduction

Cable-driven parallel robots (*CDPRs*) are a type of parallel mechanism actuated by cables instead of rigid links. With the advantages of low inertia, large potential workspace, high reconfigurability and transportability, CDPRs are widely studied and have been applied in manufacturing [1], [2], building construction [3], [4], rehabilitation [5] and rapid prototyping [6]. However, the control of CDPRs is challenging due to the unique property that cables can only apply pulling forces (*positive cable forces*). This results in a redundantly actuated system, creating challenges in the design [7], workspace analysis [8]–[12] and controller design [13]–[16] of CDPRs.

The most common type of CDPR can be regarded as the single-link cable-driven robot (SCDR), which consists 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, resulting in a large robot footprint. Moreover, SCDRs typically also have a limited range of end-effector orientations. Motivated by such limitations, multilink cable-driven robots (*MCDRs*) and hybrid cable-driven robots (*HCDRs*) have been developed.

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. Additionally, the similarities between MCDRs and anthropomorphic systems have motivated bio-robotic applications, such as exoskeletons [17], musculoskeletal arm [18] and humanoid [19] robots, and snake-like robots [20]. HCDRs are hybrid actuated systems that combine cables and direct joint actuation, and hence possess both passive and active joint degrees-of-freedom (*DoFs*). Examples of HCDRs include: the FASTKIT robot [21], where a CDPR is mounted onto multiple mobile robot platforms such that these mobile robots can enlarge its workspace and reconfigure the attachment points; and 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.

While MCDRs and HCDRs bring some very unique advantages compared with SCDRs, the control of such systems are also more complicated as the kinematics and dynamics of such CDPRs have to be described within three spaces: 1) *actuation space* refers to the control input to the robot (cable forces and command for the actuated DoFs); 2) *joint space* represents the DoFs (generalised coordinates); and the 3) *operational space* (or *task space*) denotes the pose of the end-effector. Most importantly, the dimension of the operational space is typically less than that of the generalised coordinates For example, for the SpiderArm robot the operational space may be the 6 DoF position and orientation of the end-effector while it possesses 12 DoFs in the joint space.

As a result, the operational space control problem for MCDRs and HCDRs is both important and difficult. This problem involves resolving two levels of redundancies: 1) between operational and joint spaces (*kinematic redundancy*); and 2) 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. Additionally, various constraints, such as positive cable forces, joint limits and cable interference, make the problem even more complex. Figure 1 shows the redundancy relationship between the three spaces. Due to the two-level

[1]Y. P. Chan, Y. S. Chan, and D. Lau are with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong SAR {ypchan, yschan03}@mae.cuhk.edu.hk, darwinlau@cuhk.edu.hk

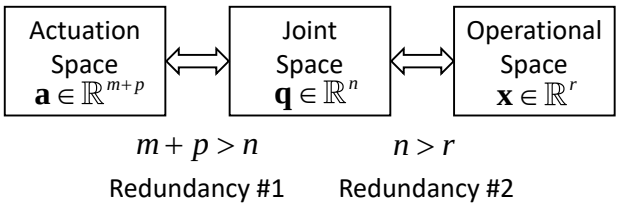


Fig. 1: Relationship between the actuation, joint and operational spaces for MCDRs and HCDRs

redundancy, existing operational space controllers developed for SCDRs [X] are not suitable for MCDRs and HCDRs.

Different methods have been studied to solve the operational space control problem for MCDRs. One common approach is the *cascading* method that considers the two levels of redundancy as two separate problems. First, a joint space trajectory that tracks the reference operational space motion is determined by solving the serial robot inverse kinematics (*serial IK*) problem. Subsequently, the cable forces required to actuate the robot to track the joint trajectory is obtained by solving the CDPR inverse dynamics (*CDPR ID*) problem [14], [22]. Within the cascading scheme, the IK can be resolved through the operational null space control developed originally for serial robots [23]. However, the joint space trajectory from the operational space control may not be feasible for CDPRs during the ID process, as the positive cable force constraint is not considered within the IK problem.

To avoid the feasibility problem of the two-stage approach, it is desired to determine both the actuation command and joint space motion in a single process within the operational space control problem. One potential approach to solve this problem is probabilistic sampling methods (PSMs), such as rapidly-exploring random trees (RRTs) [24], [25] and expansive search trees (ESTs) [26]. The control problem can then be treated as a kinodynamically constrained motion planning problem. PSMs solve for a feasible joint space trajectory that satisfies both the tracking of operational space motion and positive force constraints by probabilistically expanding a search tree. Although a feasible joint trajectory can be guaranteed with sufficient searching if such trajectory exists, the method is extremely time-consuming and must be performed offline for each trajectory. As a result, PSMs are not suitable to be used within real-time control of CDPRs.

To achieve real-time control, *direct reactive operational space control* approaches can be used to directly determine a set of feasible cable forces at each time step from an operational space trajectory. In [27], Khatib's operational space control [23] was extended to determine the muscle forces required to achieve reactive operational space control of biomechanical systems. Feasibility was addressed by incorporating the workspace of the upper arm generated from experimental human data. Since the workspace for biomechanical systems is typically regular, it can be conservatively approximated by a set of linear constraints [28]. As such, the control command can be determined by solving a convex quadratic program (*QP*). However, for CDPRs, the linear constraints approximation of the pre-generated workspace is not practical due to the high workspace irregularity and computational costs for high DoF robots. As such, [27] developed for biomechanical systems cannot be effectively used for CDPRs in general.

Another drawback of reactive control approaches is that the reactive nature only considers the instantaneous pose and does not consider the evolution of both the future joint trajectory and control input. Typically, the task performance is based on the entire trajectory (*trajectory level*), such as tracking performance or total energy, where reactive methods struggle. This drawback is particularly relevant to the operational space control of CDPRs as the performance of the later part of the trajectory is highly dependent on evolution of earlier joint motions. Furthermore, the reactive nature means that the avoidance of constraints, such as joint limits or cable interference, cannot be effectively reacted to in advance.

In summary, to the best of the authors' knowledge, there does not exist any CDPR control framework that possesses the following features: 1) perform operational space control online; 2) ensure feasibility in executing the tracking task; 3) no assumptions on the regularity of the workspace; 4) applicable to the general class of CDPRs; and 5) ability to improve trajectory-level performance over time. Such a control framework is required to conveniently and effectively operate real CDPR systems to perform various tasks in real-time.

In this paper, a generalised operational space control framework, consisting of *reactive control* and *iterative-learning control* components, is proposed for arbitrary types of CDPRs, particularly for MCDRs and HCDRs. The reactive controller solves for the actuation commands directly from the reference operational space trajectory through a convex quadratic program (*QP*) at each time step. The controller aims to achieve the tracking task, minimise control effort and avoid undesirable situations, such as the lack of manipulability and cable interference. To further improve the control performance at a trajectory-level, an iterative-learning controller (*ILC*) is proposed to improve the kinematic redundancy resolution between the operational and joint spaces after each trajectory iteration. The capabilities of the framework is illustrated through simulations using the CASPR software[1] [29] on the BMArm (MCDR), and the SpiderArm and FASTKIT HCDRs. Experiments on the BMArm are also conducted to show the capabilities of the framework to operate on hardware.

The proposed control framework possesses several unique advantages over existing methods by the use of reactive and iterative-learning controllers. First, it is capable of producing feasible solutions in real-time even on the first attempt (iteration), and then further improve its performance online over time. Second, the framework is generic where various types of objectives and constraints can be included depending on different applications. Third, the formulation can be applied to the family of CDPRs. Fourth, the ILC component effectively exploits the joint space redundancy in order to improve performance, mitigating the heavy reliance on careful and laborious

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

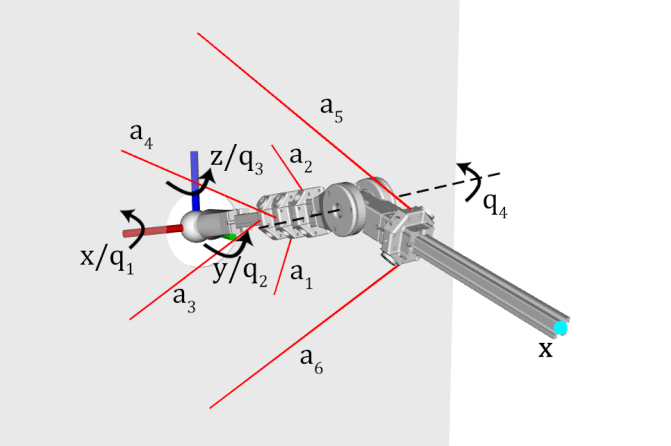


Fig. 2: BM-Arm MCDR actuated by 6 cables ($\boldsymbol{l}$: cable length; $\boldsymbol{q}$: generalised coordinates; $\boldsymbol{x}$: operational space)

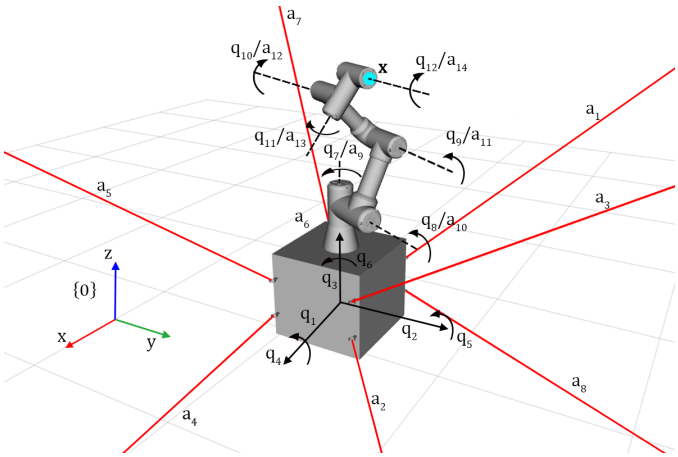


Fig. 3: SpiderArm HCDR with 4 cables and 2 active revolute joints. ($\boldsymbol{f}$: cable force; $\boldsymbol{\tau}$: active joint torque)

tuning of control gains. In summary, these features result in an easy to use framework that reduces user effort, particularly for different trajectory motions, system requirements and robots.

The remainder of the paper is organised as follows: Section II presents the generalised MCDR and HCDR models. Section III outlines the operational space control framework. Sections IV and V proposes the reactive and ILC components, respectively. Simulation and experimental results are presented in Sections VI and VII, respectively. Finally, Section VIII concludes the paper and discusses future work.

II. CDPR Modelling

A. SCDR and MCDR Model

An MCDR example is depicted in Figure 2, where the SCDR is a special case of MCDR where the number of links is one. The joint space $\boldsymbol{q} \in \mathbb{R}^n$ is defined as the generalised coordinates. The cable lengths and forces are denoted as $\boldsymbol{l} \in \mathbb{R}^m$ and $\boldsymbol{f} \in \mathbb{R}^m$, respectively. The example MCDR in Figure 2 consists of one spherical joint and one revolute joint ($n = 4$) and is actuated by 6 cables ($m = 6$). The kinematic relationship between $\boldsymbol{l}$ and $\boldsymbol{q}$ is described by

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

where $\boldsymbol{s} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is the inverse kinematics (IK) function [30]. The derivative of (1) gives

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

where $\boldsymbol{L}(\boldsymbol{q}) \in \mathbb{R}^{m \times n}$ is the joint-cable Jacobian matrix. Due to the unilateral actuation property of cables, *actuation redundancy* ($m \geq n + 1$) is needed such that the system may be capable of producing motion in all DoFs.

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 L.H.S. of the EoM (3) refers to the system wrench, which is composed of the mass-inertia matrix $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n \times n}$, the centrifugal and Coriolis vector $\boldsymbol{C}(\dot{\boldsymbol{q}}, \boldsymbol{q}) \in \mathbb{R}^n$, and the gravitational vector $\boldsymbol{G}(\boldsymbol{q}) \in \mathbb{R}^n$. 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 bounded by the minimum and maximum positive force bounds $\underline{\boldsymbol{f}}$ and $\overline{\boldsymbol{f}}$ in (4), respectively.

The operational space can be defined as the position and/or orientation of the end-effector $\boldsymbol{x} \in \mathbb{R}^r$, where r refers to the number of operational space DoFs. For example, the operational space of the robot in Figure 2 can be defined as the xyz-coordinate of the tip of the last link ($r = 3$). The kinematic relationship between the joint and operational spaces is described by

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{5}$$

where $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{r \times n}$ is the *joint-operational Jacobian* matrix. The derivative of (5) is given by

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

B. HCDR Model

HCDRs are hybrid actuated systems that combine cables and other types of actuators. For an HCDR with m-cables and p-actuators, the original CDPR joint space (now denoted as $\tilde{\boldsymbol{q}} \in \mathbb{R}^{\tilde{n}}$) is extended to include the actuated DoFs $\boldsymbol{q}_a \in \mathbb{R}^p$. As a result, the joint space of the HCDR becomes $\boldsymbol{q} = [\tilde{\boldsymbol{q}}^T \;\; \boldsymbol{q}_a^T]^T \in \mathbb{R}^n$ and $n = \tilde{n} + p$. Figure 3 shows an example of HCDR, where 4 cables are attached to a 3-DoF CDPR platform. Two links that are directly actuated by motors are further attached on the CDPR platform. The joint space of HCDRs can be defined as $\boldsymbol{q} \in \mathbb{R}^5$ and the operational space can be defined using (5) and (6) in the same way as MCDRs.

The actuation command $\boldsymbol{a}$ is formed by combining the cable force $\boldsymbol{f} \in \mathbb{R}^m$ and the active joint torque $\boldsymbol{\tau} \in \mathbb{R}^p$

$$\boldsymbol{a} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{\tau} \end{bmatrix} \in \mathbb{R}^{m+p} \tag{7}$$

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

$$\underline{\boldsymbol{a}} \leq \boldsymbol{a} \leq \overline{\boldsymbol{a}},$$
$$\underline{\boldsymbol{a}} = \begin{bmatrix} \underline{\boldsymbol{f}} \\ \underline{\boldsymbol{\tau}} \end{bmatrix}, \quad \overline{\boldsymbol{a}} = \begin{bmatrix} \overline{\boldsymbol{f}} \\ \overline{\boldsymbol{\tau}} \end{bmatrix}, \tag{8}$$

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 in (3) and (4)

$$\begin{aligned} \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\dot{\boldsymbol{q}}, \boldsymbol{q}) + \boldsymbol{G}(\boldsymbol{q}) &= \boldsymbol{N}(\boldsymbol{q})\boldsymbol{a}, \\ \boldsymbol{N}(\boldsymbol{q}) &= \begin{bmatrix} -\boldsymbol{L}(\boldsymbol{q})^T & \boldsymbol{A}(\boldsymbol{q}) \end{bmatrix}, \\ \underline{\boldsymbol{a}} \leq \boldsymbol{a} &\leq \overline{\boldsymbol{a}}, \end{aligned} \tag{9}$$

where $\boldsymbol{N}(\boldsymbol{q}) \in \mathbb{R}^{n \times (m+p)}$ is formed by the *joint-actuator* Jacobian matrix 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 (9) reduces to the EoM for MCDRs or SCDRs (3). Therefore, the extended EoM (9) will be used to describe the dynamics of CDPRs in the remainder of the paper.

III. Proposed Reactive-Iterative Operational Space Control Framework

The proposed operational space control framework consists of two primary components:

1) Reactive Quadratic Programming (RQP) controller that serves to track operational space reference trajectories in real-time (Section IV).
2) Iterative-Learning Controller (*ILC*) that improves the task performance through iterations when the trajectory is repeatedly executed (Section V).

Figure 4 shows the structure of the proposed framework. The reactive control component performs the tracking task with the proposed QP controller, which solves both the kinematic and actuation redundancies simultaneously in a single process. At each time step of the operational space reference trajectory, defined by $\mathcal{X}_d(t) = \{\boldsymbol{x}_d(t), \dot{\boldsymbol{x}}_d(t), \ddot{\boldsymbol{x}}_d(t)\}$, a convex QP problem is solved to determine a set of feasible actuation command $\boldsymbol{a}$ for the CDPR to track the reference trajectory.

Exploiting the kinematic redundancy from (6), a set of *null space exploration parameters* $\boldsymbol{u} \in \mathbb{R}^n$ is proposed. This allows different joint behaviours that can produce the same desired operational space motion to be explored and optimised by the ILC to improve the performance of the reactive controller when the same motion is repeated. After completing each trajectory (defined as *iteration*), the task performance is quantified by an objective function $P(\boldsymbol{u})$ that considers the norms of actuation commands, tracking error and other objectives.

The ILC can be solved as an optimisation problem with $\boldsymbol{u}$ as the optimising variable. With iterative optimisation methods, $\boldsymbol{u}$ is updated after each iteration and is fed back to the RQP controller. The characteristic of the controller in resolving the kinematic redundancy is hence altered, and such change is expected to result in improvements of $P(\boldsymbol{u})$. As a result, the proposed controller essentially operates at two time-scales: 1) the reactive controller acts at each instantaneous point in time (*fast time-scale*); and 2) the ILC operates at each trajectory repetition (*slower time-scale*).

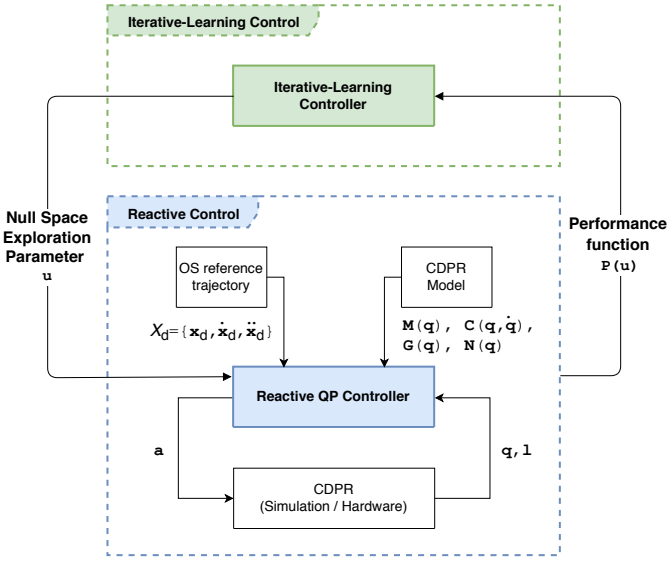


Fig. 4: Operational Space Control Framework

IV. Reactive Quadratic Programming Controller

The main objective of the RQP controller is to determine a set of feasible actuation commands $\boldsymbol{a}$ that actuates the system to track the operational space trajectory $\mathcal{X}_d$, while avoiding undesirable poses, at every time instant. To achieve this, the following convex QP formulation is proposed such that both levels of redundancy (kinematic and actuation) are solved within a single problem

$$\begin{aligned} &\min_{\ddot{\boldsymbol{q}}, \boldsymbol{a}} V(\mathcal{X}_d, \ddot{\boldsymbol{q}}, \boldsymbol{a}) \\ \text{subject to } &\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\dot{\boldsymbol{q}}, \boldsymbol{q}) + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{N}(\boldsymbol{q})\boldsymbol{a}, \\ &\underline{\boldsymbol{a}} \leq \boldsymbol{a} \leq \overline{\boldsymbol{a}}, \\ &\boldsymbol{\Phi} \begin{bmatrix} \ddot{\boldsymbol{q}} \\ \boldsymbol{a} \end{bmatrix} \leq \boldsymbol{\rho}, \end{aligned} \tag{10}$$

where $V(\mathcal{X}_d, \ddot{\boldsymbol{q}}, \boldsymbol{a})$ is the controller's objective function, and $\boldsymbol{\Phi}$ and $\boldsymbol{\rho}$ define the inequality constraints for the controller.

The objective function $V(\mathcal{X}_d, \ddot{\boldsymbol{q}}, \boldsymbol{a})$ is comprised of three quadratic components: 1) *tracking function* $g_t(\mathcal{X}_d, \ddot{\boldsymbol{q}})$ resolving the kinematic redundancy; 2) *actuation function* $g_f(\boldsymbol{a})$ resolving the actuation redundancy and minimising control effort, and; 3) *avoidance function* $g_a(\ddot{\boldsymbol{q}})$ avoiding undesirable situations. Hence, $V(\mathcal{X}_d, \ddot{\boldsymbol{q}}, \boldsymbol{a})$ can be expressed as

$$V(\mathcal{X}_d, \ddot{\boldsymbol{q}}, \boldsymbol{a}) = g_t(\mathcal{X}_d, \ddot{\boldsymbol{q}}) + \alpha \cdot g_f(\boldsymbol{a}) + \beta \cdot g_a(\ddot{\boldsymbol{q}}), \tag{11}$$

where $\alpha > 0$ and $\beta > 0$ are scaling factors for the actuation function and avoidance function, respectively. As a general guideline, α should be selected as $\alpha \ll 1$ due to the lower priority of control effort minimisation compared with tracking, while $\alpha \leq \beta \leq 1$.

The RQP controller contains three types of linear constraints: the EoM (9), actuation bounds and other task constraints. The EoM ensures that the system dynamics can be satisfied, while the actuation bounds ensure the resulting actuation commands are feasible. Other inequality constraints are hard constraints that maintain the system's capability of

achieving the tracking task and is detailed in Section IV-C.

A. Operational Space Tracking

To determine the joint acceleration $\ddot{\boldsymbol{q}}$ to achieve operational space tracking, the kinematic redundancy between $\ddot{\boldsymbol{q}}$ and the reference operational space trajectory $\mathcal{X}_d$ has to be resolved. Hence, the tracking function $g_t(\mathcal{X}_d, \ddot{\boldsymbol{q}})$ is designed to minimise the difference between $\ddot{\boldsymbol{q}}$ and a reference tracking joint acceleration while resolving the kinematic redundancy

$$g_t(\mathcal{X}_d, \ddot{\boldsymbol{q}}) = \left\| \ddot{\boldsymbol{q}} - \boldsymbol{J}_W^{\dagger} \boldsymbol{b}(\mathcal{X}_d) \right\|_2^2, \tag{12}$$

where

$$\boldsymbol{J}_W^{\dagger} = \boldsymbol{W}_q^{-1} \boldsymbol{J}^T (\boldsymbol{J} \boldsymbol{W}_q^{-1} \boldsymbol{J}^T)^{-1}, \tag{13}$$

$$\boldsymbol{b}(\mathcal{X}_d) = \ddot{\boldsymbol{x}}_d - \dot{\boldsymbol{J}}\dot{\boldsymbol{q}} - \boldsymbol{K}_d(\boldsymbol{J}\dot{\boldsymbol{q}} - \dot{\boldsymbol{x}}_d) - \boldsymbol{K}_p(\boldsymbol{x} - \boldsymbol{x}_d), \tag{14}$$

where $\boldsymbol{K}_d, \boldsymbol{K}_p \in \mathbb{R}^{r \times r}$ are diagonal matrices with positive elements, $\boldsymbol{J}_W^{\dagger}$ is the weighted pseudoinverse of $\boldsymbol{J}$ [31] 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.

When the kinematic relationships (5) and (6) are substituted into (12) and $g_t(\mathcal{X}_d, \ddot{\boldsymbol{q}})$ is minimised, (13) and (14) result in the operational space error dynamics

$$\ddot{\boldsymbol{e}} + \boldsymbol{K}_d \dot{\boldsymbol{e}} + \boldsymbol{K}_p \boldsymbol{e} = \boldsymbol{0}_r, \tag{15}$$

where $\boldsymbol{e} = \boldsymbol{x} - \boldsymbol{x}_d \in \mathbb{R}^r$ is the operational space error and $\boldsymbol{0}_r \in \mathbb{R}^r$ is a zero column vector.

B. 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 (9); 2) be within the feasible actuation bound (8); and 3) minimise some desired objectives such as minimal *control effort*. The EoM and actuation bounds are represented as constraints in the QP (10), and the minimisation of control effort can be simply expressed by the objective function component

$$g_f(\boldsymbol{a}) = \boldsymbol{a}^T \boldsymbol{W}_a \boldsymbol{a}, \tag{16}$$

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

To demonstrate the capability of resolving the kinematic and actuation redundancies with the tracking function $g_t(\mathcal{X}_d, \ddot{\boldsymbol{q}})$ and the actuation function $g_f(\boldsymbol{a})$, simulations on a 2-link spherical-revolute MCDR - *BMArm* (Figure 5) are performed. The joint space $\boldsymbol{q} \in \mathbb{R}^4$, 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, and is actuated by 6 cables, $\boldsymbol{a} \in \mathbb{R}^6$, as shown in Figure 5. The operational space $\boldsymbol{x} \in \mathbb{R}^3$ is defined as the xyz-coordinate of the second link's tip in frame $\{0\}$.

As an example, a star-shaped trajectory on the xz-plane with edge length of 0.15m and starting point of $\boldsymbol{x}_d = [0, 0.595, 0]^T$ (Figure 5) is tracked with $\boldsymbol{K}_p = 200\boldsymbol{I}_3$, $\boldsymbol{K}_d = 28\boldsymbol{I}_3$ (damping ratio ≈ 1), $\alpha = 1 \times 10^{-6}$, $\boldsymbol{W}_q = \boldsymbol{I}_4$ and $\boldsymbol{W}_a = \boldsymbol{I}_6$, where

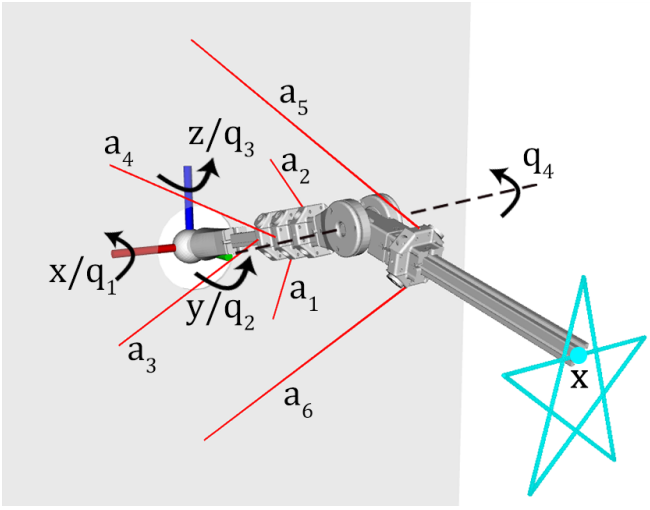


Fig. 5: Schematic diagram of BMArm (2-link spherical-revolute MCDR)

$\boldsymbol{I}_n$ refers to a n-by-n identity matrix. The initial joint position is set at $\boldsymbol{q} = [0.18, 0, 0, -0.45]^T$. Figure 6a shows that the RQP controller is capable of tracking 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 $\boldsymbol{a}$ in Figures 6b and 6c, respectively.

C. 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 RQP controller only consists of the tracking and actuation functions. This is since the controller is incapable of preventing the system from encountering undesirable situations, such as loss of manipulability, cable interference, and reaching mechanical joint limits. For example, consider the simulation of the star trajectory in Section IV-B but with a different initial joint position $\boldsymbol{q} = [0.18, 0.2, 0, -0.45]^T$. The results in Figure 7 show that the operational space tracking becomes unstable from approximately 4.2 seconds.

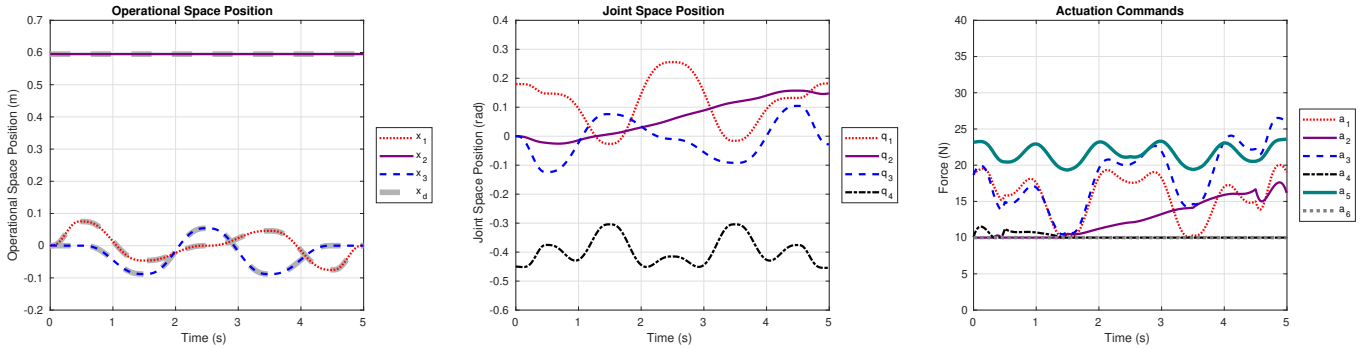
The divergence is caused by the system's lack of manipulability $K(\boldsymbol{q})$, as defined using *unilateral dexterity* [32]

$$K(\boldsymbol{q}) = \frac{\sigma_n(\boldsymbol{L}(\boldsymbol{q}))}{\sigma_1(\boldsymbol{L}(\boldsymbol{q}))} \frac{\sqrt{m+1}\,\eta_{min}(\boldsymbol{q})}{\sqrt{\eta_{min}^2(\boldsymbol{q}) + 1}}, \tag{17}$$

where σ_1 and σ_n refers 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 normalised null space vector $\hat{\boldsymbol{n}}(\boldsymbol{q})$, which is obtained by projecting the vector $\mathbb{1} = [1, 1, ..., 1]^T \in \mathbb{R}^m$ into the null space of $\boldsymbol{L}(\boldsymbol{q})$

$$\boldsymbol{n}(\boldsymbol{q}) = (\boldsymbol{I} - \boldsymbol{L}(\boldsymbol{q})\boldsymbol{L}^{\dagger}(\boldsymbol{q}))\mathbb{1}, \tag{18}$$

where $\hat{\boldsymbol{n}}(\boldsymbol{q}) = \frac{\boldsymbol{n}(\boldsymbol{q})}{\|\boldsymbol{n}(\boldsymbol{q})\|}$. Consequently, $K(\boldsymbol{q})$ reaches a maximum value of 1 when the system is *unilateral isotropic* [32], and reaches 0 when the system loses its capability to generate any arbitrary direction of wrench through positive cable forces. The manipulability metric $K(\boldsymbol{q})$ along the star-shaped trajectory is depicted in Figure 7c. It can be seen that K



(a) Operational space $\mathbf{x}(t)$ (reference in grey) (b) Joint space motion $\mathbf{q}(t)$ (c) Cable force profiles $\mathbf{a}(t)$

Fig. 6: BMArm simulation tracking star-shaped trajectory (without avoidance function), with initial $\boldsymbol{q} = [0.18, 0, 0, -0.45]^T$.

reaches 0 after around 3.5 seconds, meaning that the system loses its capability to generate any direction of wrench and leads to the divergence behaviour of the controller.

To avoid such problems, it is proposed to use a combination of hard and soft constraints within the RQP controller from (10) 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. Without loss of generality, it will be assumed that the function $h(\boldsymbol{q}, \dot{\boldsymbol{q}})$ represents one of the measures of desirable scenarios, and the following can be achieved:

- Satisfaction of $h(\boldsymbol{q}, \dot{\boldsymbol{q}}) \geq h_{min}$ as a hard constraint
- Increase of $h(\boldsymbol{q}, \dot{\boldsymbol{q}})$ as an objective to avoid h_{min}

1) Hard linear constraints: The system can be regarded as being close to undesirable situations when $h_{min} \leq h(\boldsymbol{q}, \dot{\boldsymbol{q}}) \leq h_{min} + \Delta_h$, where $\Delta_h \in \mathbb{R}^+$ represents the region size that is considered to be close to the undesirability. Since $h(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is typically non-linear, such as in (17), it is proposed to add a velocity level constraint when $h_{min} \leq h(\boldsymbol{q}, \dot{\boldsymbol{q}}) \leq h_{min} + \Delta_h$

$$\dot{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = \frac{\partial h(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\partial \boldsymbol{q}} \dot{\boldsymbol{q}} + \frac{\partial h(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\partial \dot{\boldsymbol{q}}} \ddot{\boldsymbol{q}} \geq \epsilon, \tag{19}$$

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

$$\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}_p + \ddot{\boldsymbol{q}} \delta t, \tag{20}$$

where δt is the time step and $\dot{\boldsymbol{q}}_p$ is the joint space velocity at the previous time step. By combining (19) and (20), the following linear constraint can be included in the reactive QP controller to avoid undesirable situations

$$\boldsymbol{\Phi} = \left[-\delta t \frac{\partial h}{\partial \boldsymbol{q}} - \frac{\partial h}{\partial \dot{\boldsymbol{q}}} \quad \mathbf{0}_{m+p} \right], \quad \boldsymbol{\rho} = \frac{\partial h}{\partial \boldsymbol{q}} \dot{\boldsymbol{q}}_p - \epsilon \ , \tag{21}$$

where $\mathbf{0}_{m+p}$ is an $m+p$ zero row vector. Since $\frac{\partial h(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\partial \boldsymbol{q}}$ and $\frac{\partial h(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\partial \dot{\boldsymbol{q}}}$ are both constant row vectors at a particular state $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$, (21) is a linear constraint within the RQP controller.

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 create sudden changes of acceleration, which potentially creates non-smooth joint space motions. Hence, in addition to hard constraints, an avoidance function $g_a(\ddot{\boldsymbol{q}})$ is proposed to more smoothly guide the system away from undesirable situations prior to reaching the hard constraints, increasing the capability to avoid undesirable situations in advance. The avoidance function $g_a(\ddot{\boldsymbol{q}})$ can be formulated as

$$g_a(\ddot{\boldsymbol{q}}) = \left\| \ddot{\boldsymbol{q}} - \hat{\ddot{\boldsymbol{q}}}_A \right\|_2^2, \tag{22}$$

where $\hat{\ddot{\boldsymbol{q}}}_A$ refers to the *avoiding acceleration*, which is designed to drive the system away from undesirable situations.

The avoidance objective can be formulated as a desired velocity in order to increase the gradient of function $h(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and the desired acceleration $\hat{\ddot{\boldsymbol{q}}}_A$ expressed as

$$\hat{\ddot{\boldsymbol{q}}}_A = k_1 \frac{\partial h}{\partial \dot{\boldsymbol{q}}}^T + \frac{k_2}{\delta t} \left(\frac{\partial h}{\partial \boldsymbol{q}}^T - \dot{\boldsymbol{q}}_p \right), \tag{23}$$

where $k_1, k_2 > 0$ are constants governing the strength related to joint position or velocity level avoidance, respectively.

In cases where more than one type of undesirable situations have to be avoided, $\hat{\ddot{\boldsymbol{q}}}_A$ can be defined as the linear combination of N joint accelerations

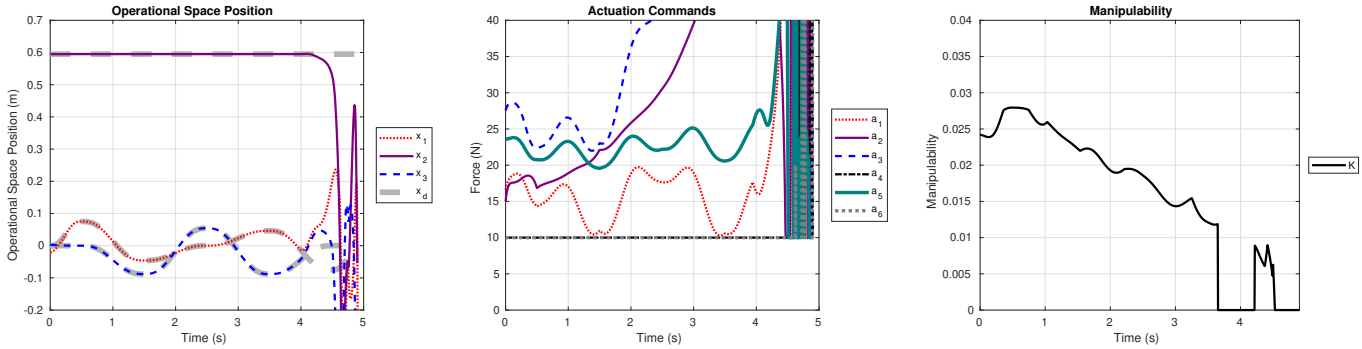
$$\hat{\ddot{\boldsymbol{q}}}_A = \sum_{i=1}^{N} w_i(\boldsymbol{q}, \dot{\boldsymbol{q}}) \cdot \ddot{\boldsymbol{q}}_{a,i}, \tag{24}$$

where $\ddot{\boldsymbol{q}}_{a,i}$ corresponds to the joint acceleration required to avoid the i-th undesirable situations (23). The avoidance of multiple situations can be prioritised with weights $w_i > 0$, where $\sum_{i=1}^{N} w_i(\boldsymbol{q}, \dot{\boldsymbol{q}}) = 1$ which are dependent on the current states $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ of the system.

Figure 8 demonstrates the effect of the avoidance of low manipulability by setting $h(\boldsymbol{q}, \dot{\boldsymbol{q}}) = K(\boldsymbol{q})$ for the same task as in Figure 7, with $\beta = 0.01$. It can be seen from Figures 8a and 8b that the tracking task is completed without divergence and significant decreases in manipulability (Figure 8c compared with 7c). This demonstrates the ability of the avoidance function to maintain the system's manipulability to complete the tracking task.

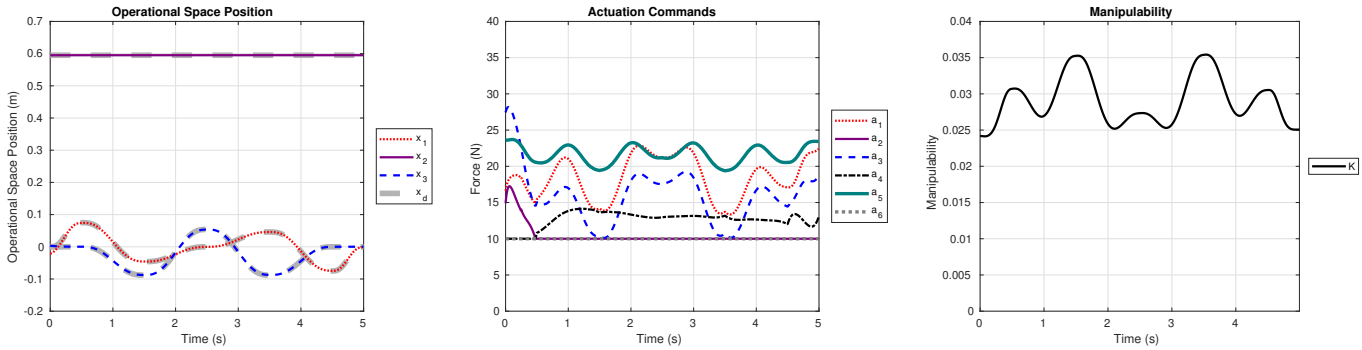
V. Iterative-Learning Controller

Iterative-learning control (ILC) is a well-studied control method [33], [34] based on the idea that, given a repeatable



(a) Operational space $\mathbf{x}(t)$ (reference in grey) (b) Cable force profiles $\mathbf{a}(t)$ (c) Manipulability $K(\boldsymbol{q})$

Fig. 7: BMArm simulation tracking star-shaped trajectory (without avoidance function), with initial $\boldsymbol{q} = [0.18, 0.2, 0, -0.45]^T$.



(a) Operational space $\mathbf{x}(t)$ (reference in grey) (b) Cable force profiles $\mathbf{a}(t)$ (c) Manipulability $K(\boldsymbol{q})$

Fig. 8: BMArm simulation of star-shaped trajectory with avoidance function ($\beta = 0.01$), with initial $\boldsymbol{q} = [0.18, 0.2, 0, -0.45]^T$.

task with the same initial conditions, information in a certain *iteration* can be used to improve the task performance in the next iteration [35]. Traditionally, ILC updates a set of *time-wise* control inputs over a finite time interval with a designed learning rule, such that the system achieves perfect tracking and the control input converges to the optimal control input over iterations [33].

For the application to CDPRs, the control input would naturally be the cable forces $\boldsymbol{f}(t)$. However, due to the cable force constraints and actuation redundancy, existing ILCs cannot be directly applied onto CDPRs to determine the cable forces. This is due to the lack of closed-form relationship between the control inputs and system states [36], [37], as demonstrated in Section IV by the need of optimisation-based controllers. As such, numerical gradient or search-based methods are required in order to determine the optimal control input [38]. However, given the high dimensions of the problem to determine the set of input states at every trajectory time instant, it is not possible to achieve real-time control using ILC directly.

To overcome this, an ILC is to be used in combination with the RQP controller presented in Section IV. Rather than iteratively learning the cable force command for all time instants, which can be a large set of parameters with constraints, it is proposed to learn a set of *null space exploration parameters* $\boldsymbol{u} \in \mathbb{R}^{n+2}$ within the RQP controller. These parameters are constant for an entire trajectory and are designed to affect the characteristic of the controller in resolving the operational to joint space kinematic redundancy. As a result, by updating $\boldsymbol{u}$, the joint space profile to achieve better performance when executing the desired operational space trajectory will be explored. Furthermore, compared with applying ILC on cable forces, the null space exploration parameters are unconstrained and hence can be solved more easily.

The proposed framework enables the separation of the responsibilities and time-scale of the reactive and ILC components. The reactive controller is responsible for maintaining system stability and satisfying the unilateral constraints of $\boldsymbol{a}$ when encountering new trajectories without any past experiences. The ILC focuses on improving the performance through exploring the joint space redundancy when the operational space task is repeated. The reactive controller operates real-time at each time instant (*faster-time scale*). On the other hand, the ILC performs each update at the end of each trajectory cycle (*iteration*) at a *slower-time scale*.

Mathematically, the ILC can be solved as an unconstrained nonlinear optimisation problem

$$\boldsymbol{u}^\star = \arg\min_{\boldsymbol{u}} P(\boldsymbol{u}), \tag{25}$$

where $P(\boldsymbol{u})$ is the trajectory performance and $\boldsymbol{u}^\star$ is the optimal set of null space parameters. As the unconstrained optimisation problem (25) is only performed once for each trajectory cycle, a large variety of iterative optimisation methods can be used to improve $P(\boldsymbol{u})$ in the next iteration.

A. Trajectory-Based Performance Function

The evaluation of the trajectory performance, such as tracking accuracy and actuation efforts, is done after each trajectory iteration. In the proposed ILC, at least two quantities should be considered in the evaluation: *tracking accuracy* and *actuation effort*. As different quantities usually vary in scale and dimension, the following performance function is proposed

$$P(\boldsymbol{u}) = \|\boldsymbol{E}(\boldsymbol{u})\|_F \, \|\boldsymbol{A}(\boldsymbol{u})\|_F \prod_i \|\boldsymbol{R}_i(\boldsymbol{u})\|_F \,, \tag{26}$$

where $\|\cdot\|_F$ refers to the Frobenius norm. The components of $P(\boldsymbol{u})$ can be described by:

- $\boldsymbol{E}(\boldsymbol{u}) = [\boldsymbol{e}_1, ..., \boldsymbol{e}_T]$ is the vector of operational space tracking error for the sample time steps $t = [t_1, ..., t_T]$ of the trajectory and $\boldsymbol{e}_j = \boldsymbol{x}(t_j) - \boldsymbol{x}_d(t_j)$.
- $\boldsymbol{A}(\boldsymbol{u}) = [\boldsymbol{a}_1, ..., \boldsymbol{a}_T]$ is the vector of actuation effort for the sample time steps $t = [t_1, ..., t_T]$ of the trajectory where $\boldsymbol{a}_j \equiv \boldsymbol{a}(t_j)$.
- Other performance functions, such as manipulability or tension factor, can also be represented by functions $\boldsymbol{R}_i(\boldsymbol{u}) = [\boldsymbol{r}_1, ..., \boldsymbol{r}_T]$ for time steps $t = [t_1, ..., t_T]$

B. Null Space Exploration Parameters

In order to identify the joint space behavior that results in the best task performance $P(\boldsymbol{u})$, a set of null space exploration parameters $\boldsymbol{u}$ is designed to explore the null space of the joint-operational space Jacobian $\boldsymbol{J}$ in the iteration (trajectory) domain. This exploration is achieved by extending the tracking function (12) from the reactive controller, such that $g_t(\mathcal{X}_d, \ddot{\boldsymbol{q}})$ is also dependent on $\boldsymbol{u}$ and can be defined as

$$g_t(\mathcal{X}_d, \ddot{\boldsymbol{q}}, \boldsymbol{u}) = \left\| \ddot{\boldsymbol{q}} - \hat{\ddot{\boldsymbol{q}}}_T(\mathcal{X}_d, \boldsymbol{u}) \right\|_2^2, \tag{27}$$

$$\hat{\ddot{\boldsymbol{q}}}_T(\mathcal{X}_d, \boldsymbol{u}) = \boldsymbol{J}_W^\dagger \boldsymbol{b}(\mathcal{X}_d) + (\boldsymbol{I} - \boldsymbol{J}_W^\dagger \boldsymbol{J}) \hat{\ddot{\boldsymbol{q}}}_0(\boldsymbol{u}), \tag{28}$$

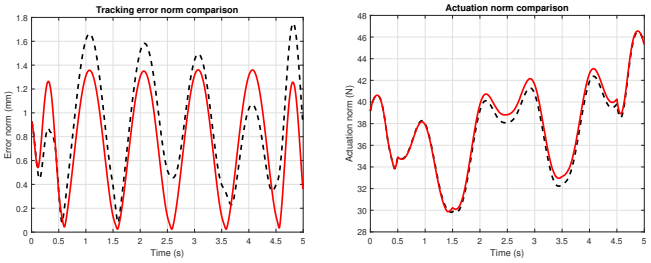
where $\hat{\ddot{\boldsymbol{q}}}_T$ is obtained by summing the weighted least-square solution $\boldsymbol{J}_W^\dagger \boldsymbol{b}$ with the projection of a reference joint space acceleration vector $\hat{\ddot{\boldsymbol{q}}}_0(\boldsymbol{u})$ in the null space of $\boldsymbol{J}$. In contrast to reactive null-space projection methods that aim to optimise certain objective function at each time step [23], [39], the projected reference acceleration $\hat{\ddot{\boldsymbol{q}}}_0(\boldsymbol{u})$ is updated by the ILC through the iterations (slower time-scale).

As actuation constraints are handled by the reactive QP controller, the selection of $\hat{\ddot{\boldsymbol{q}}}_0(\boldsymbol{u})$ can focus on exploring the kinematic redundancy to improve the controller's performance. In the proposed controller, $\hat{\ddot{\boldsymbol{q}}}_0(\boldsymbol{u})$ is formulated as

$$\hat{\ddot{\boldsymbol{q}}}_0(\boldsymbol{u}) = \boldsymbol{D}(\boldsymbol{u}) \boldsymbol{J}_W^\dagger \boldsymbol{b}, \tag{29}$$

where $\boldsymbol{D}(\boldsymbol{u}) \in \mathbb{R}^{n \times n}$ is a diagonal matrix that scales the weighted least-square solution $\boldsymbol{J}_W^\dagger \boldsymbol{b}$. Note that it is not necessary for $\boldsymbol{D}(\boldsymbol{u})$ to be positive definite. The objective of such choice is to provide a null space deviation with reference to the weighted least-square solution, where the scale of the deviation is dependent on $\boldsymbol{u}$, and hence controlled by the ILC.

Additionally, from the example in Section IV-C, it is shown that the task performance is affected by the inclusion of avoidance functions. As such, the scaling factors α and β in (11) would affect the joint space behavior and hence the task performance. Figure 9 shows an example of the effect of α in performing the star-shaped trajectory twice, with the same initial pose $\boldsymbol{q} = [0.18, 0, 0, -0.45]^T$. Scaling factors of $\alpha = 1 \times 10^{-2}$ and $\alpha = 1 \times 10^{-6}$, both without avoidance function, show differences in both the tracking error and actuator effort.



(a) Norm of tracking error (b) Norm of actuation command

Fig. 9: Comparison between $\alpha = 1 \times 10^{-2}$ (black dashed) and $\alpha = 1 \times 10^{-6}$ (red solid) on BMArm.

Algorithm 1 Proposed Control framework using Pattern Search for ILC

Input: $\mathcal{X}_d$ - operational space reference trajectory
Input: $\boldsymbol{u}_0$ - initial null space exploration parameters
Input: δ_u - step size;
Output: $\boldsymbol{u}^\star$ - optimal null space exploration parameters
Output: $\boldsymbol{Q}^\star$ - optimal trajectory of joint space position
Output: $\boldsymbol{A}^\star$ - optimal trajectory of actuation commands

$P^\star \leftarrow \infty$ (initialise $P^\star$)
for i-th iteration of ILC **do**
  // Explore neighborhood of $\boldsymbol{u}_i$
  for j-th dimension of $\boldsymbol{u}_i$ **do**
    $\overline{u}_{i,j} \leftarrow u_{i,j} + \delta_u$ (update of $\boldsymbol{u}$)
    // Execute trajectory with updated $\boldsymbol{u}$
    $\boldsymbol{Q}, \boldsymbol{A} \leftarrow$ Algorithm 2 with $\mathcal{X}_d$, $\boldsymbol{u}$ as inputs
    $P(\boldsymbol{u}) \leftarrow$ (26) using $\boldsymbol{Q}(\overline{\boldsymbol{u}}_{i,j})$ and $\boldsymbol{A}(\overline{\boldsymbol{u}}_{i,j})$
  end for
  $P_i^\star \leftarrow$ best P in the i-th iteration
  $\boldsymbol{u}_i^\star \leftarrow$ corresponding $\boldsymbol{u}$ to $P_i^\star$
  if $P_i^\star < P^\star$ **then**
    $\boldsymbol{u}^\star \leftarrow \boldsymbol{u}_i^\star$;
    $\boldsymbol{Q}^\star \leftarrow \boldsymbol{Q}(\boldsymbol{u}^\star)$;
    $\boldsymbol{A}^\star \leftarrow \boldsymbol{A}(\boldsymbol{u}^\star)$;
  else
    $\delta_u \leftarrow \frac{\delta_u}{2}$
  end if
end for
return $\boldsymbol{u}^\star, \boldsymbol{Q}^\star, \boldsymbol{A}^\star$

Considering these factors that affect the task performance, the null space exploration parameter set is proposed to be

$$\boldsymbol{u} = [d_1, ..., d_n, \ln(\alpha), \ln(\beta)]^T, \tag{30}$$

where $d_1, ..., d_n$ are diagonal elements of $\boldsymbol{D}$, while $\ln(\alpha)$ and $\ln(\beta)$ ensures that $\alpha, \beta > 0$ for unconstrained choices of $\boldsymbol{u}$.

Algorithm 2 Reactive QP Controller for a single trajectory

Input: $\mathcal{X}_d$ - operational space reference trajectory
Input: $\boldsymbol{u}$ - null space exploration parameters
Output: $\boldsymbol{Q}$ - trajectory of joint space position
Output: $\boldsymbol{A}$ - trajectory of actuation commands
1: **for** $i \leftarrow 1, \ldots, T$ **do**
2: $\quad \boldsymbol{q}_i, \dot{\boldsymbol{q}}_i \leftarrow$ from robot state
3: $\quad g_t(\mathcal{X}_d, \ddot{\boldsymbol{q}}, \boldsymbol{u}) \leftarrow$ (27) and (29)
4: $\quad g_f \leftarrow$ (16)
5: $\quad g_a \leftarrow$ (22) - (24)
6: $\quad \boldsymbol{\Phi}, \boldsymbol{\rho} \leftarrow$ (21)
7: $\quad \ddot{\boldsymbol{q}}_i, \boldsymbol{a}_i \leftarrow$ solving (10) using $\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i, g_t, g_f, g_a, \boldsymbol{\Phi}, \boldsymbol{\rho}$
8: $\quad$ Use $\ddot{\boldsymbol{q}}_i, \boldsymbol{a}_i$ to execute on the robot
9: $\quad$ Add $\boldsymbol{q}_i$ to $\boldsymbol{Q}$
10: $\quad$ Add $\boldsymbol{a}_i$ to $\boldsymbol{A}$
11: **end for**
12: **return** $\boldsymbol{Q}, \boldsymbol{A}$

C. Optimisation Method

Although the null space exploration parameters provide a great potential in exploiting the kinematic redundancy and bringing improvements in task performance, the complex relationship between these parameters and the performance function makes the manual tuning of these parameters extremely difficult. As observed from (25) and (26), the ILC can be solved as an unconstrained nonlinear optimisation problem through a great variety of optimisation methods. As the choice of $P(\boldsymbol{u})$ can potentially result in irregular terrains, gradient-based optimisation methods such as BFGS [40] may not be suitable for the problem. Instead, search-based methods such as Pattern Search [41], Nelder-Mead simplex method [42], Particle Swarm Optimisation (PSO) [43] and Covariance Matrix Adaption Evolution Strategies (CMA-ES) [44] are recommended to more effectively determine $\boldsymbol{u}$.

Algorithm 1 illustrates the way in which the ILC operates with pattern search as the optimisation method. Furthermore, it also demonstrates how the ILC and the reactive QP controller (Algorithm 2) interact with each other. With the generalisabilty of the framework, other optimisation methods can be easily implemented by changing the initialisation procedure and the update method of $\boldsymbol{u}$, lines 1 and 5 of Algorithm 1, respectively.

VI. Simulation Results

To further demonstrate the capability and features of the proposed framework, simulation results on two HCDRs: *SpiderArm* and *FASTKIT* are presented. All simulations were performed with the open-source Cable-robot Analysis and Simulation Platform for Research (CASPR) [29] software using a computer with an Intel Core i7-6700 CPU @ 3.40GHz and 16.0GB of RAM, using MATLAB R2018a (64-bit). The QP in the reactive controller was solved using qpOASES [45].

A. SpiderArm

The SpiderArm is an HCDR that combines a 6-DoF robot arm on the end-effector of a spatial SCDR (Figure 3). In addition to the large translational workspace of SCDR, the installation of the robot arm leads 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 8 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 orientation of the SCDR in frame $\{0\}$, and the joints $q_7, q_8, \ldots, q_{12}$ refer to the joint angles of the robot arm. The operational space was 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 $\{0\}$.

Two types of undesirable situations were considered for the SpiderArm: loss of manipulability of the SCDR and cable interference. To avoid these situations, the avoiding acceleration $\hat{\ddot{\boldsymbol{q}}}_A$ was designed as

$$\hat{\ddot{\boldsymbol{q}}}_A = w(\boldsymbol{q})\ddot{\boldsymbol{q}}_K + (1 - w(\boldsymbol{q}))\ddot{\boldsymbol{q}}_D, \tag{31}$$

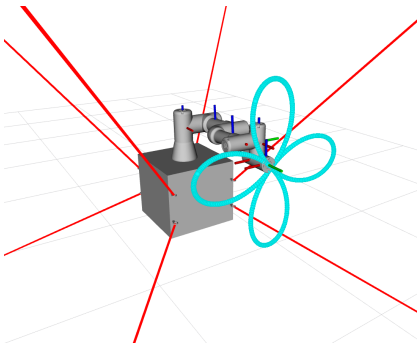
$$w(\boldsymbol{q}) = \begin{cases} 0.5, \ \exists \ i \ \mid \ \delta_i(\boldsymbol{q}) < \epsilon, \\ 1, \ \textit{otherwise}, \end{cases} \tag{32}$$

where $w(\boldsymbol{q})$ is a weighting function that prioritises the two components of $\hat{\ddot{\boldsymbol{q}}}_A$: 1) the acceleration that increases manipulability of the SCDR $\ddot{\boldsymbol{q}}_K$, and; 2) the acceleration that avoids cable interference $\ddot{\boldsymbol{q}}_D$. The distance between each cable and each rigid link of robot arm pair (*cable-link pair*) was calculated numerically by treating cables and links as line segments [46], denoted as $\boldsymbol{\delta}_i(\boldsymbol{q})$. In SpiderArm, up to $\boldsymbol{i} = 48$ cable-link pairs were considered each time step. When one of the distance was less than the user-defined buffer ϵ, $\ddot{\boldsymbol{q}}_K$ and $\ddot{\boldsymbol{q}}_D$ shared the same level of emphasis. Otherwise, the controller focused on maintaining the manipulability of the SCDR.

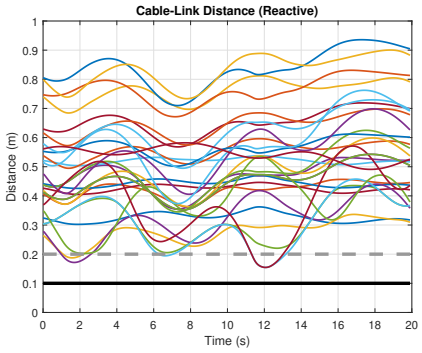
The avoiding acceleration for loss of manipulability $\ddot{\boldsymbol{q}}_K$ was calculated in a similar manner as in Section IV-C2. For the avoidance of cable interference, $\ddot{\boldsymbol{q}}_D$ was obtained by setting $h(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \delta_{min}(\boldsymbol{q})$, where $\delta_{min}(\boldsymbol{q})$ is the distance between the closest cable-link pair at $\boldsymbol{q}$. By applying the avoiding acceleration $\ddot{\boldsymbol{q}}_D$, the distance between the closest cable-link pair should increase. Instead of handling multiple cable-link pairs, only the worst case was handled, as the solution space for an avoiding acceleration that increases multiple cable-link distances is very limited.

1) Reactive Controller: A flower-shaped trajectory (Figure 10a) was tracked to demonstrate the capability of the proposed reactive controller. The trajectory lasted for 20 seconds, with an amplitude of 0.45m and a center at $[2, 1.1, 1.14]^T$. Reference orientation for the end-effector was set at $[0, 0, -\frac{\pi}{2}]^T$ throughout the trajectory. The control frequency was set at 20Hz, and the tracking proportional gains for position and orientation were set at $40\boldsymbol{I}_3$ and $5\boldsymbol{I}_3$, respectively. Derivative gains were set at $12.6\boldsymbol{I}_3$ and $4.5\boldsymbol{I}_3$ for position and orientation, respectively, such that the damping ratio equals to 1.

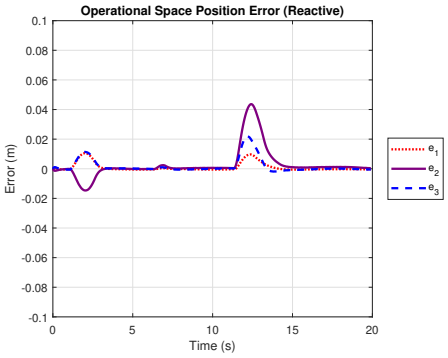
One of the main challenges in this tracking task lies in the avoidance of cable interference. Due to the large amplitude of the trajectory (0.45m), there is a high chance that the robot arm hits the cables. Hence, it is necessary to include the avoidance function. The buffer ϵ was set at 0.2m, and the hard limit was



(a) Flower-shaped trajectory on the yz-plane

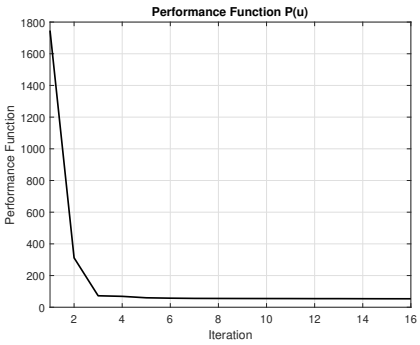


(b) Distances between cables and links $\boldsymbol{\delta}(\boldsymbol{q})$ (Buffer: grey dashed; Limit: black solid)

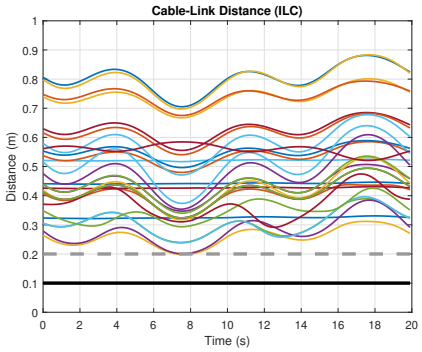


(c) Operational space position error $\boldsymbol{e}(t)$

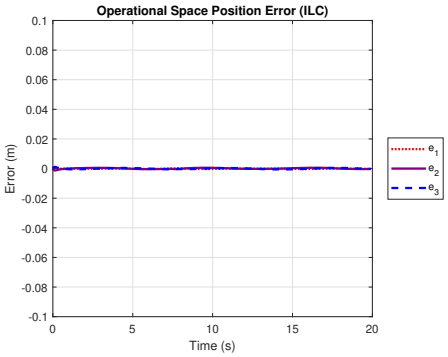
Fig. 10: Simulation of tracking task on the SpiderArm with the reactive controller.



(a) Performance function $P(\boldsymbol{u})$ over the 15 iterations



(b) Distances between cables and links $\boldsymbol{\delta}(\boldsymbol{q})$ (Buffer: grey dashed; Limit: black solid)



(c) Operational space position error $\boldsymbol{e}(t)$

Fig. 11: ILC performed on the SpiderArm with results after 15 iterations.

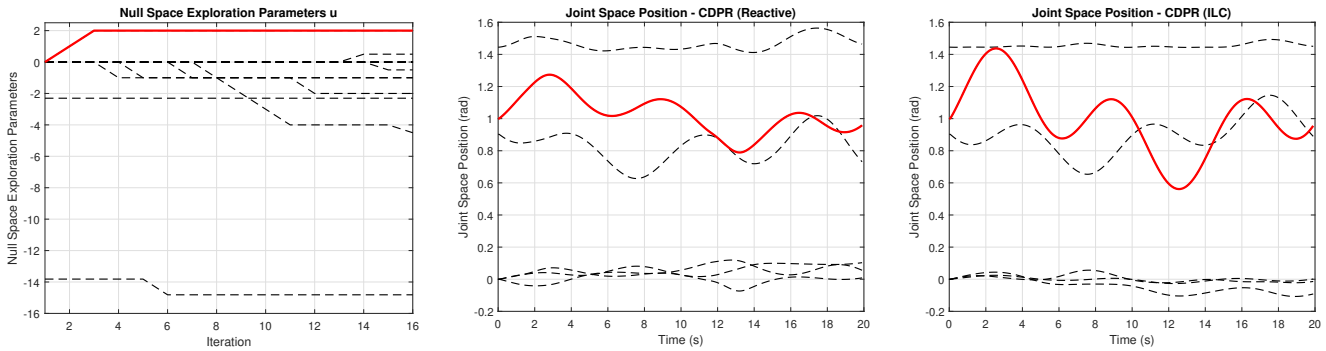
set at 0.1m, which stopped the system once encountered. The scaling factors for actuation function α and avoidance function β were set at 1×10^{-6} and 0.01, respectively.

Figure 10b shows the distances between cables and links during the tracking task. At approximately 2 and 12 seconds, it can be seen that the distance between a cable-link pair dropped below the buffer of 0.2m (grey dashed line). 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 buffer region. However, due to the weighted formulation of the QP objective function, the need for avoidance led to reduced emphasis on tracking. As a result, it can be seen from Figure 10c that at the corresponding time when the avoidance function was activated, the tracking error experienced an increase with an amplitude up to 4cm.

2) Iterative-Learning Controller (ILC): In order to improve the tracking performance of the reactive controller, the null space of the system was explored by the proposed ILC. The pattern search (algorithm 1) was used with an initial step size of 1, the performance function $P(\boldsymbol{u})$ (26) was optimised with the initial null space exploration parameters $\boldsymbol{u}$ set at $[0, ...0, \ln(1 \times 10^{-6}), \ln(0.01)]^T$. For $\boldsymbol{u} \in \mathbb{R}^{14}$, the pattern search algorithm has to perform $14 \times 2 + 1 = 29$ evaluations in each iteration. Hence, for a total of 15 iterations, $29 \times 15 = 435$ evaluations were performed.

Figure 11a shows the performance function $P(\boldsymbol{u})$ in the ILC process. It can be seen that P was reduced from approximately 1700 to below 100. The reason for such significant improvement is that the ILC discovered a joint space trajectory that does not cause cable interference. From Figure 11b, it is shown that the distances between cable-link pairs stayed above the buffer (grey dashed) throughout the entire trajectory. As a result, the avoidance function was not activated during the task, and the RQP controller emphasised more on the tracking task. Hence, the tracking error maintained at a low level (Figure 11c) and resulted in a low value of $P(\boldsymbol{u})$. This demonstrates that the ILC has the potential to produce a motion that both decreases operational space error and can avoid interference between the robot arm and cables, through exploiting the joint space redundancy throughout the trajectory.

In addition to the improvement in performance, further insights regarding the task can also be obtained from the null space exploration parameters. It can be observed from Figure 11a that the performance function experienced the greatest decrease in the first 2 iterations. By referring to the change in $\boldsymbol{u}$ (Figure 12a), such improvement corresponded to the increase in u_2 (solid red line), which is related to the importance of y movements of the SCDR when the kinematic redundancy is resolved. As a result, the amplitude of y movements of the SCDR (red solid) increased after performing ILC (Figure 12b-12c). These results demonstrate that changes in $\boldsymbol{u}$ can affect the joint space motion of the robot, and are capable of



(a) Evolution of null space exploration parameters $\boldsymbol{u}$ over iterations (u_2 is highlighted in red)

(b) Joint space position of SCDR before ILC (y: red solid)

(c) Joint space position of SCDR after ILC (y: red solid)

Fig. 12: Effect of increased u_2 on the y-position of SCDR after ILC.

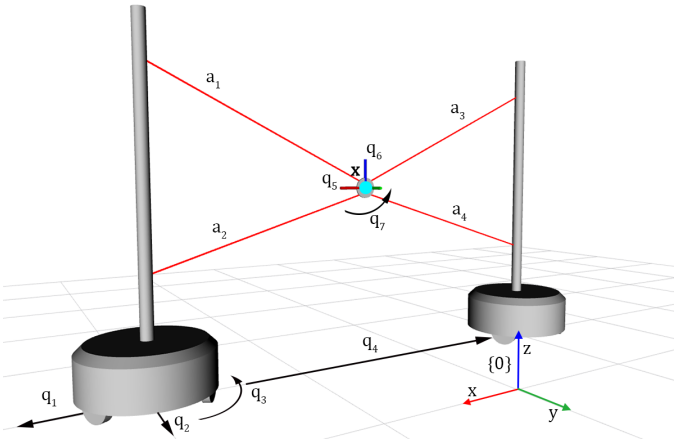


Fig. 13: Schematic diagram of FASTKIT-Planar

improving task performance.

B. FASTKIT-Planar

The FASTKIT-Planar robot is the planar version of the FASTKIT HCDR in [21], which consists of a planar CDPR between two mobile bases (Figure 13). The joint space of the robot is defined as $\boldsymbol{q} = [q_1, q_2, ..., q_7]^T$, where q_1 and q_2 refers to the xy-coordinates of the first mobile base, and q_3 refers to its orientation. The distance between the two mobile bases is modelled by a prismatic joint, denoted as q_4. For the planar robot, q_5, q_6 and q_7 represents the xy-coordinates and the orientation of the end-effector in the frame of the first mobile base. The planar robot is actuated by 4 cables a_1, a_2, a_3 and a_4, with 2 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 was 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 $\{0\}$.

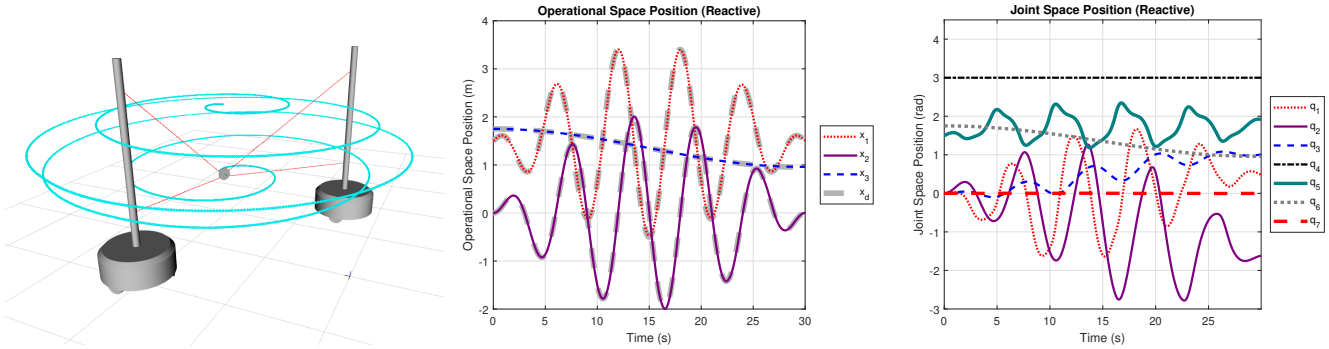
1) Reactive Control: The design of the FASTKIT-Planar robot leads to several challenges in reactive control. First, the mobile bases have unlimited workspace, and their movements can possibly have no effect on the end-effector. Second, the controller should keep the movement of the planar CDPR between the two mobile bases. This example shows the proposed reactive controller's ability to be applied to very different systems with no controller tuning required. Moreover, the QP formulation (10) allows the resolution of kinematic and actuation redundancies, even for a system with high number of redundant DoFs. Additionally, the planar robot should not collide with the mobile bases. This was achieved with an avoidance function, by setting $h(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \delta_{min}(\boldsymbol{q})$, where $\delta_{min}(\boldsymbol{q})$ refers to the minimium distance between the planar end-effector and the closest mobile base.

A helix-shaped trajectory with horizontal radius of 2m, vertical height of 0.8m and center at $[1.5, 0, 1.35]^T$ was tracked with 100Hz (Figure 14a). Simple choices were made for the parameters of the controller, with $\boldsymbol{K}_p = 200\boldsymbol{I}_3$, $\boldsymbol{K}_d = 28\boldsymbol{I}_3$ (damping ratio ≈ 1), $\alpha = 1 \times 10^{-6}$ and $\beta = 0.1$. From Figure 14b, it can be seen that the operational space trajectory was successfully tracked, with the peak error amplitude less than 0.1m. The resulting joint position $\boldsymbol{q}$ is shown in Figure 14c. Joints that do not affect the end-effector, such as q_4 and q_7, were resolved by the controller to produce minimal amounts of motion. For the x-position of the planar robot q_5, which could possibly collide with the mobile bases, was pushed back when it encounters the buffer value of 2m. The total computational time needed for the calculation of system dynamics and the reactive QP process had an average of 1.42ms and a worst-case of 3.63ms, which showed that the controller is capable of achieving online control.

2) Iterative-Learning Control (ILC): The kinematic redundancy on the FASTKIT-Planar robot was explored and exploited with the use of the ILC. With the same parameters used in the SpiderArm case, the proposed ILC formulation was directly applied on the FASTKIT-Planar. After running 15 iterations, the performance function decreased from 87089 to 5007 with a similar $P(\mathbf{u})$ performance profile as in Figure 11a. Figures 15a to 15c show that the tracking error, mobile base actuation norm and cable force norm all decreased after ILC. These results not only demonstrate the capability of the ILC to improve performance, but also the generalisability of the proposed framework to different robots.

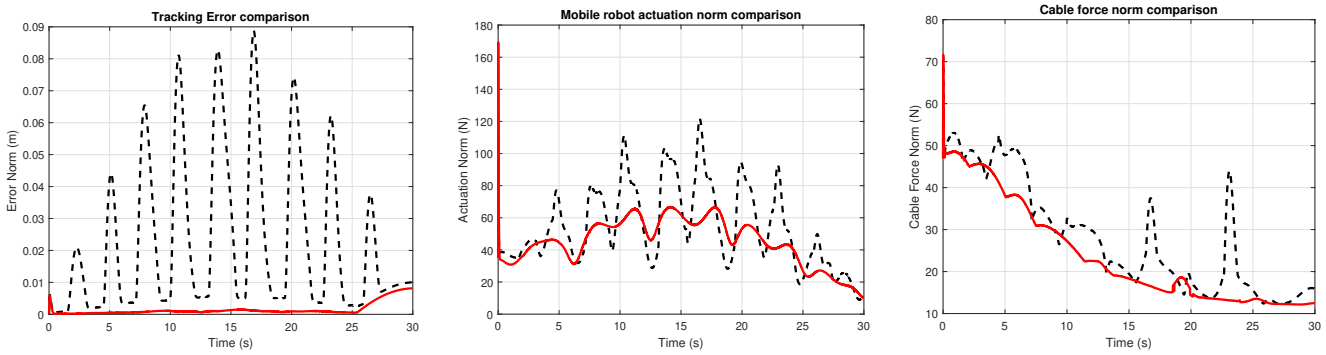
VII. Hardware Results

The proposed control framework was applied on the BMArm robot hardware (Figure 16). The hardware was



(a) Helix-shaped trajectory (b) Operational space position $\boldsymbol{x}(t)$ (c) Joint space position $\boldsymbol{q}(t)$

Fig. 14: Simulation results on FASTKIT-Planar using only reactive QP controller.



(a) Tracking error $\mathbf{e}(t)$ (b) Mobile base actuation norm (c) Cable force norm

Fig. 15: Comparison results after performing ILC on the FASTKIT-Planar (Before ILC: black dashed; After ILC: red solid).

controlled by two computers: 1) a 64-bit computer, with a 3.4GHz Core i7-6700 processor and 16GB ram, running the CASPR-ROS software platform [47] responsible for the low-level motor control; and 2) a 64-bit computer, with a 2.8GHz Core i7-7700HQ processor and 16GB ram, running CASPR in MATLAB for the proposed reactive and ILC framework.

Cable force commands were determined by the reactive controller, and the corresponding cable lengths were calculated through forward dynamics. Length commands were sent to CASPR-ROS at 200Hz and cable lengths were controlled by myomuscle units [48]. Joint space position $\boldsymbol{q}$ and velocity $\dot{\boldsymbol{q}}$ were determined through forward kinematics (FK) using the cable length feedback. Operational space position $\boldsymbol{x}$ and velocity $\dot{\boldsymbol{x}}$ were determined through forward kinematics (FK) with $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$. Finally, load cells were placed in-line with the cables to measure cable forces $\boldsymbol{f}$.

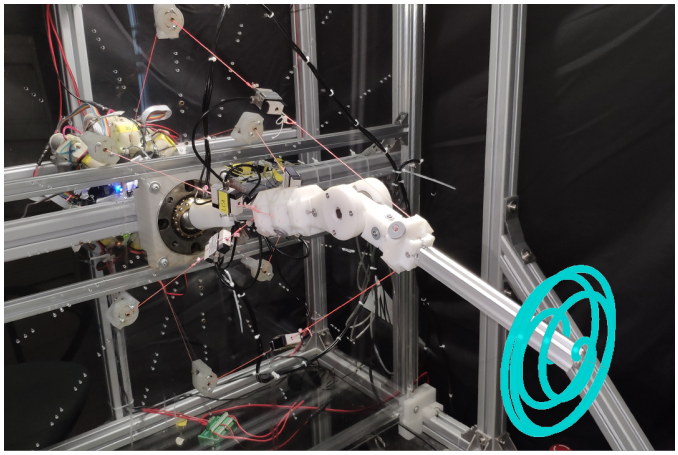
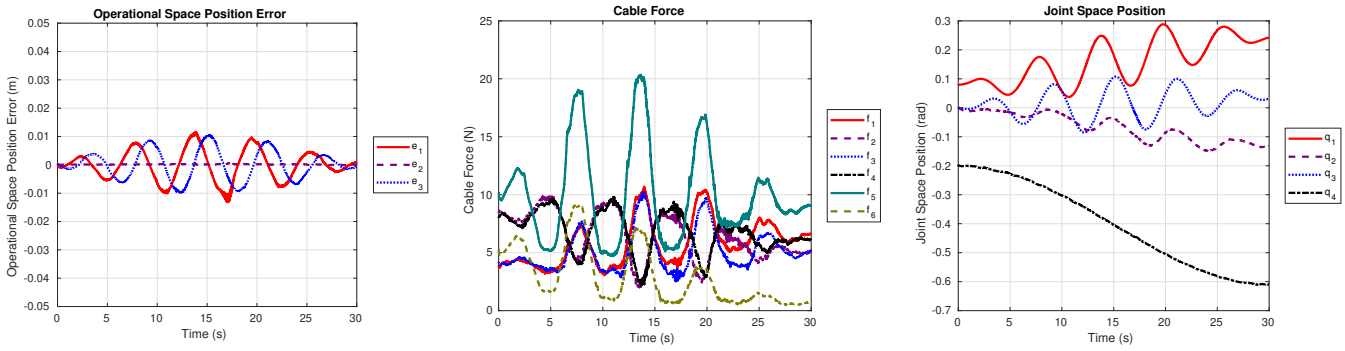


Fig. 16: Physical implementation of the BMArm and the helix-shaped trajectory

A. Reactive Control

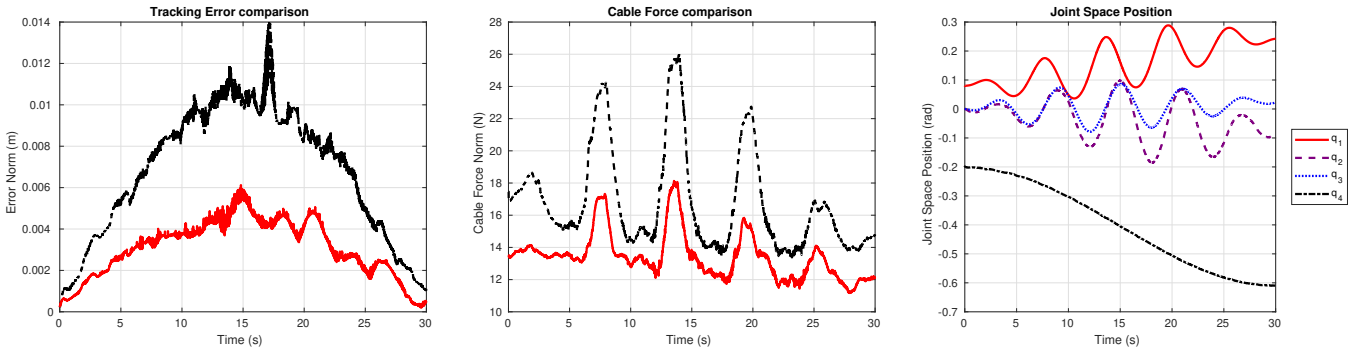
A helix-shaped trajectory similar to that in Section VI-B was tracked, which rotated about the y-axis with a radius of 0.06m, thickness of 0.012m, and center at $[0, 0.595, 0]^T$. The starting joint pose was set at $\boldsymbol{q} = [0.08, 0, 0, -0.2]^T$. Control parameters were set at $\boldsymbol{K}_p = 20000\boldsymbol{I}_3$, $\boldsymbol{K}_d = 282\boldsymbol{I}_3$ (damping ratio ≈ 1), $\alpha = 1 \times 10^{-6}$, $\beta = 1$. 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.

Figure 17 shows the results of the hardware tracking task. The tracking error in the operational space of amplitude lower than 0.015m was obtained (Figure 17a). Cable forces measured by the load cells (Figure 17b) also shows that the tracking task was achieved with positive cable forces. The kinematic redundancy is resolved by the reactive controller resulted in the joint space trajectory shown in Figure 17c. For the computational time of the reactive controller, a worst-case of 15.5ms was recorded during the warm-up stage of the QP



(a) Operational space position error $\boldsymbol{e}(t)$ (b) Cable forces measured by load cells $\boldsymbol{f}(t)$ (c) Joint space position $\boldsymbol{q}(t)$

Fig. 17: Helix trajectory tracked on the BMArm hardware using only reactive QP controller.



(a) Tracking error norm (b) Cable force norm measured from load cells (c) Joint space position $\boldsymbol{q}$ after ILC

Fig. 18: Comparison results after performing ILC on the BMArm hardware (Before ILC: black dashed; After ILC: red solid).

process. However, 99% of the time steps achieved an average of 2.2ms, which demonstrates the capability of the proposed reactive QP controller to achieve operational space control of CDPR hardware online in real-time. These behaviours were consistent with that observed when performing simulations for the BMArm.

B. Iterative-Learning Control (ILC)

ILC was also performed on the BMArm hardware using the proposed formulation. Due to the uncertainties on hardware such as friction and sensor noise, the performance evaluation may vary for the same set of control parameters. Therefore, instead of pattern search used in the simulations in Section VI, Particle Swarm Optimisation (PSO) method was used in the ILC of hardware, as it is more robust against the potentially inconsistent performance evaluation. For the PSO, 6 particles were used to explore the solution space of $\boldsymbol{u}$ over 15 iterations, resulting in a total of $6\times15 = 90$ evaluations. Position bounds for the particles were set at $\boldsymbol{u}_{min} = [-2,-2,-2,-2,\ln(1\times10^{-7},\ln(0.1)]^T$, $\boldsymbol{u}_{max} = [2,2,2,2,\ln(1\times10^{-5},\ln(10)]^T$. The inertia weight ω was set at 0.73, while particle and swarm best parameters ϕ_p, ϕ_g were both set at 1.5 [43].

Results after performing ILC are shown in Figure 18. Comparison results show that the tracking has improved, with the maximum amplitude of the error norm reducing from 0.014m to around 0.006m (Figure 18a). Figure 18b also shows that the cable forces used in the tracking task was reduced along the entire trajectory. Both improvements in the tracking error and cable force norm contributed to a significant reduction of the performance function, from 765 to 272.

Furthermore, by comparing the joint space trajectory before (Figure 17c) and after ILC (Figure 18c), it can be observed that such improvement is closely related to an increase in the movement of q_2. This corresponds to the twisting motion along the y-axis. Although excessive twisting creates potential risk for loss of manipulability (Section IV-C), the ILC was capable of determining 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 for updating $\boldsymbol{u}$ and evaluating $P(\boldsymbol{u})$ after completing each trajectory, has insignificant values of 0.75ms and 16.71ms, respectively. This implies that without a time-consuming planning process before performing each new task, the proposed reactive and iterative-learning control framework was capable of completing a new task and improving the task performance online on robot hardware.

VIII. Conclusion

An operational space control framework that consists of reactive and iterative-learning control features was proposed for CDPRs. The reactive controller solved a convex QP problem such that an operational space reference trajectory can be tracked with feasible cable forces, while avoiding undesirable situations. The iterative-learning control component further improved the performance of the reactive controller at a trajectory level, with the use of a set of null space

exploration parameters. To demonstrate the capability and generalisability of the proposed framework, simulations were performed on 2 HCDRs, the SpiderArm and FASTKIT-Planar robot. Experimental results on a 2-link MCDR, BMArm, were also presented to demonstrate the practicality of the framework in controlling hardware online. Future work will focus on the experience transfer between ILC results of different tracking tasks, such that faster convergence can be potentially obtained when encountering new tasks.

ACKNOWLEDGEMENT

The work was supported by the Research Grants Council (Early Career Scheme Reference No. 24200516).

REFERENCES

[1] J. S. Albus, R. V. Bostelman, and N. Dagalakis, "The NIST robocrane," *J. Robot. Syst.*, 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 modelling 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 II, L. S. Bryson, and D. Castro-Lacouture, "Cable-suspended robotic contour crafting system," *Autom. Constr.*, vol. 17, no. 1, pp. 45–55, 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. Sim. Model. Program. Auton. Robot.*, 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, 2012.

[6] J.-B. Izard, A. Dubor, P.-E. Hervé, E. Cabay, D. Culla, M. Rodriguez, and M. Barrado, "Large-scale 3d printing with cable-driven parallel robots," *Constr. Robots.*, 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. Springer London, 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, pp. 011 010/1–10, 2010.

[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, 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, pp. 032 303/1–7, 2016.

[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, 2011.

[12] S. Rezazadeh and S. Behzadipour, "Workspace analysis of multibody cable-driven mechanisms," *J. Mech. Robot.*, vol. 3, no. 2, pp. 021 005/1–10, 2011.

[13] A. B. Alp and S. K. Agrawal, "Cable suspended robots: Design, planning and control," in *Proc. IEEE Int. Conf. Robot. Autom.*, 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, 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, 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, 2008.

[17] S. K. Agrawal, V. N. Dubey, J. J. Gangloff, Jr., E. Brackbill, and V. Sangwan, "Optimization and design of a cable driven upper arm exoskeleton," in *Proc. ASME Int. Des. Eng. Tech. Conf. and Comp. Inf. Eng. Conf.*, 2009, pp. 86 516/1–8.

[18] G. Yang, S. K. Mustafa, C. B. Pham, and S. H. Yeo, "Kinematic design of a 7-DOF cable-driven humanoid arm: a solution-in-nature approach," in *Proc. IEEE/ASME Int. Conf. Adv. Int. Mechatronics*, 2005, pp. 444–449.

[19] 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.*, 2011, pp. 3029–3034.

[20] P. Racioppo and P. Ben-Tzvi, "Modeling and control of a cable driven modular snake robot," in *Proc. IEEE Int. Conf. Contr. Technol. Applications*, 2017, pp. 468–473.

[21] 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*, ser. Mechanisms and Machine Science, C. Gosselin, P. Cardou, T. Bruckmann, and A. Pott, Eds. Springer International Publishing, 2018, vol. 53, pp. 268–279.

[22] 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, 2015.

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

[24] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Computer Science Department, Iowa State University, Tech. Rep., 1998.

[25] S. M. LaValle and J. James J. Kuffner, "Randomized kinodynamic planning," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, 2001.

[26] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *Int. J. Robot. Res.*, vol. 21, no. 3, pp. 233–255, 2002.

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

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

[29] 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. Robot. Syst.*, 2016, pp. 3004–3011.

[30] 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, 2013.

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

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

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

[34] 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, 2006.

[35] J.-X. Xu and Y. Tan, *Linear and Nonlinear Iterative Learning Control*, ser. Lecture Notes in Control and Information Sciences. Springer-Verlag Berlin Heidelberg, 2003, vol. 291.

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

[37] 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, 2013.

[38] 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, 1991.

[39] V. D. 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, 2008.

[40] 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.

[41] R. Hooke and T. A. Jeeves, ""direct search" solution of numerical and statistical problems," *J. ACM*, vol. 8, no. 2, pp. 212–229, 1961.

[42] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 7, no. 4, pp. 308–313, 1965.

[43] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. Sixth Int. Sypm. Micro Mach. Human Sci.*, 1995, pp. 39–43.

[44] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation," in *Proc. IEEE Int. Conf. Evol. Comput.*, 1996, pp. 312–317.

[45] 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.

[46] V. J. Lumelsky, "On fast computation of distance between line segments," *Inform. Process. Lett.*, vol. 21, pp. 55–61, 1985.

[47] 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*, ser. Mechanisms and Machine Science, C. Gosselin, P. Cardou, T. Bruckmann, and A. Pott, Eds., vol. 53. Springer International Publishing, 2018, pp. 50–61.

[48] H. G. Marques, C. Maufroy, A. Lenz, K. Dalamagkidis, U. Culha, M. Siee, P. Bremner, and the MYOROBOTICS Project Team, "MYOROBOTICS: a modular toolkit for legged locomotion research using musculoskeletal designs," in *Proc. 6th International Symposium on Adaptive Motion of Animals and Machines (AMAM'13)*, 2013.