Decentralized Ability-Aware Adaptive Control for Multi-robot Collaborative Manipulation

Lei Yan, Theodoros Stouraitis, and Sethu Vijayakumar

Abstract—Multi-robot teams can achieve more dexterous, complex and heavier payload tasks than a single robot, yet effective collaboration is required. Multi-robot collaboration is extremely challenging due to the different kinematic and dynamics capabilities of the robots, the (enforced) limited communication between them, and the uncertainty of the system parameters. In this paper, a Decentralized Ability-Aware Adaptive Control (DA^3C) is proposed to address these challenges based on two key features. Firstly, the common manipulation task is represented using the proposed nominal task ellipsoid, which is used to maximize each robot's force capability online by optimizing its configuration. Secondly, a decentralized adaptive controller is designed to be Lyapunov stable in spite of significant heterogeneous actuation constraints of the robots and uncertain physical parameters of the object and environment. In the proposed framework, decentralized coordination and load distribution between the robots is achieved without communication, while only the control deficiency is broadcast if any of the robots reaches its force limits. In this case, the object's reference trajectory is modified in a decentralized manner to guarantee stable interaction. Finally, we perform several numerical and physical simulations to analyse and verify the proposed method with heterogeneous multi-robot teams in collaborative manipulation tasks.

Index Terms—Distributed robot systems, redundant robots, manipulation planning, motion control, robust/adaptive control.

I. INTRODUCTION

Collaboration with other agents can often be beneficial. For example, a multi-robot team like the one shown in Fig. 1 is more dexterous and robust in heavy and large object manipulation tasks [1] than a single robot. Also, in human-robot collaboration scenarios [2], the human's input can improve the intelligence and adaptability of the team. Yet, collaboration is not trivial, due to the effects of one agent's actions on the planning, control and decision of others.

Here, we investigate multi-robot collaborative manipulation tasks, where a decentralized robot team needs to achieve a common objective, while each robot has different motion and force capabilities. To perform such collaborative tasks,
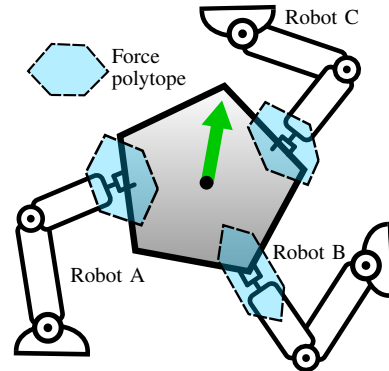
Fig. 1: Pictorial description of the multi-robot collaborative manipulation setup, where the ability of each robot is illustrated as a force polytope.

each robot within the team should maximize its contribution to the task, appropriately distribute the load among other robots, and adapt its behaviour according to the capabilities of the other robots in the team. Traditional centralized control methods have been used for multi-robot collaboration, but assume access to an accurate model of the robot team and full observability of the state of the other robots and the object. However, when considering the characteristics of real-world multi-robot teams [3], the most pressing problems are: (i) heterogeneity of the robots' capabilities, (ii) uncertainty of the system's physical parameters and (iii) lack of high bandwidth communication between the robots.

Therefore, we propose a method to maximize the force capability of each robot while designing a decentralized adaptive controller. Using this framework, we achieve the shared manipulation task under modelling uncertainties, input constraints and band-limited communication.

Motion and force capabilities: A manipulability metric [4] was first proposed as a measure of the capability of robotic mechanisms, and has been broadly used for redundancy control [5]. Utilizing the task-oriented manipulability measure, the optimal joint configuration of a redundant manipulator can be determined [6]. An efficient closed-form calculation of the task space manipulability was presented for a 7-DOF manipulator [7]. For a multiple-arm system, the task-space force and velocity manipulability ellipsoids were given in [8]. Based on the study of the weighting matrix for dynamic manipulability of robots, two physically meaningful choices for weighting matrix were provided [9]. To simplify the calculation of the dynamic manipulability, the weighted manipulability ellipsoid can be used to approximate the manipulability polytope [10]. In this paper, the weighted force manipulability ellipsoid (WFME) will be adopted to optimize the force polytope of
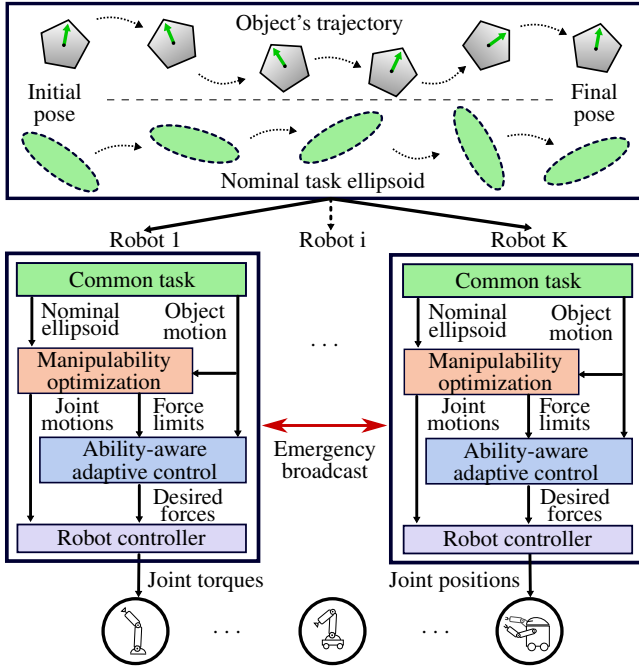
Fig. 2: Flowchart of decentralized ability-aware adaptive control.

the manipulator.

Redundancy exploitation: For dexterous manipulation [11], [12], redundant manipulators were adopted to enlarge workspace, avoid singularities and collisions with the environment. Further, manipulator's redundancy can also be explored to maximize manipulability [13]. The stiffness feasibility regions of redundant manipulators and a global task-oriented stiffness optimization were used to select robot poses [14] and configurations [15] for force and stiffness control, respectively. This highlights the important role of the robot's configuration in interaction tasks. The configuration of a redundant robot was also optimized while manipulating an object in space, to minimize disturbances on the base [16]. Geometry-aware methods were used to provide a manipulability-based redundancy resolution [17], [18], which enabled tracking of manipulability ellipsoids. Yet, the desired ellipsoid is either pre-defined by a human expert or pre-recorded from demonstrations. Here, we define a nominal task ellipsoid which is generated from the manipulation task automatically and present a task-oriented manipulability optimization to match the WFME of the robot with the nominal task ellipsoid.

Decentralized adaptive control: A decentralized model reference adaptive controller was proposed to deal with uncertain physical parameters of the collaborative multi-robot system [19]. For manipulation tasks with inaccurate kinematic model, the adaptive controller was used to handle the closed kinematic chain constraint and achieve accurate motion tracking with minimum-norm actuation force [20]. Recently, Deep Neural Networks were adopted to model system uncertainties in model reference adaptive control [21]. In studies of multiple mobile manipulators, distributed coordination control and synchronous cooperation control were presented to deal with time delays and switching topologies [22]. A distributed cooperation scheme was adopted for networked mobile ma-

nipulators, which exploits the formation-based task allocation and task-oriented strategy [23]. A distributed impedance controller has also been used for collaborative manipulation with event-triggered communication [24]. Recently, a decentralized adaptive controller [1] for multiple collaborative mobile robots was introduced. This controller can track the reference velocity trajectory without a priori knowledge of the agent's position and payload properties. Yet, all adaptive controllers described above did not consider force and torque constraints (manipulability) of the robots.

Considering a multi-robot collaborative manipulation task, a decentralized ability-aware adaptive control (DA^3C) framework (shown in Fig. 2) is proposed. Our method can handle both uncertain system parameters and input constraints without full communication between the robots. In the investigated multi-robot collaborative manipulation setup each robot has access to the desired manipulation task, which is described as a nominal task ellipsoid. Each robot tracks the nominal task ellipsoid using the task-oriented manipulability optimization method, while the DA^3C enables multi-robot coordination with respect to the common manipulation task. This paper's main contributions are summarised as follows:

1) A nominal task ellipsoid is defined based on the common manipulation task, and it is used to optimize the force capability of each manipulator.
2) A decentralized adaptive controller under input constraints is designed and proven to be Lyapunov stable.
3) Different heterogeneous multi-robot systems with input and communication constraints realize collaborative manipulation tasks using the proposed decentralized ability-aware adaptive control that guarantees stability and convergence.

The remainder of this paper is organized as follows. The preliminary work is presented in Section II. In Section III, we define the nominal task ellipsoid and present the task-oriented manipulability optimization. The decentralized ability-aware adaptive control is described in Section IV. In depth analysis of the proposed method is carried out in Section V, where several numerical and physical simulations of collaborative manipulation tasks are performed. The conclusion and future work are discussed in Section VI.

II. PRELIMINARIES

A. Manipulability Ellipsoid and Force Polytope

Generally speaking, the force manipulability ellipsoid can be used to approximately describe the force capability of the manipulator. In the simplest case, if we consider an arbitrary n-DOF manipulator robot k with the same torque limit across all its joints, we can obtain the force manipulability ellipsoid [4] using joint torque $\boldsymbol{\tau}_k$ such that $\|\boldsymbol{\tau}_k\|_2 \leqslant 1$. This ellipsoid is a subset of all realizable forces and is defined as

$$\boldsymbol{F}_k^T \left(\boldsymbol{J}_k \boldsymbol{J}_k^T \right) \boldsymbol{F}_k \leqslant 1, \tag{1}$$

where $\boldsymbol{J}_k$ is the Jacobian matrix of robot k and $\boldsymbol{F}_k$ is the force (and torque) at the end-effector of robot k.

However, typically manipulators have different torque limits for each joint, expressed as $|\tau_k^i| \leqslant \tau_{k,max}^i$ for $i = 1, ..., n$,

where $\tau_{k,max}^i$ is the maximum torque of i-th joint of robot k. Thus, the force polytope [10] of the manipulator is described by $2n$ bounding inequalities as

$$- \boldsymbol{\tau}_{k,max} \leqslant \boldsymbol{J}_k^T \boldsymbol{F}_k \leqslant \boldsymbol{\tau}_{k,max}. \qquad (2)$$

As shown on the left side of Fig. 3, the force polytope corresponding to joint torque limits in (2) can be approximated by the weighted force manipulability ellipsoid (WFME) as

$$\boldsymbol{F}_k^T \boldsymbol{J}_k \boldsymbol{W}_k^T \boldsymbol{W}_k \boldsymbol{J}_k^T \boldsymbol{F}_k \leqslant 1, \qquad (3)$$

where $\boldsymbol{W}_k = \text{diag}\left(\frac{1}{\tau_{k,max}^1}, \cdots, \frac{1}{\tau_{k,max}^n}\right)$ is a weighting matrix used to formulate the WFME.

B. Object's Dynamics

For multiple robot collaborative manipulation scenarios, the equation of motion of the object can be written as

$$\begin{bmatrix} m_o \dot{\boldsymbol{v}}_o \\ \boldsymbol{I}_o \dot{\boldsymbol{\omega}}_o + \boldsymbol{\omega}_o \times (\boldsymbol{I}_o \boldsymbol{\omega}_o) \end{bmatrix} = \boldsymbol{F}_t + \boldsymbol{F}_f + \boldsymbol{G}_o, \qquad (4)$$

where m_o and $\boldsymbol{I}_o$ are the mass and inertia of the target object, $\boldsymbol{v}_o$ and $\boldsymbol{\omega}_o$ are respectively the linear velocity and angular velocity of the object, $\boldsymbol{G}_o = [m_o \boldsymbol{g}^T, \boldsymbol{0}]^T$ is the gravity, $\boldsymbol{F}_f$ is the linear and rotational friction force which is modelled as $\boldsymbol{F}_f = [\boldsymbol{f}_l^T, \boldsymbol{f}_r^T]^T = [-\mu_l \boldsymbol{v}_t^T, -\mu_r \boldsymbol{\omega}_t^T]^T$, μ_l and μ_r are the linear and rotational friction coefficients, respectively. $\boldsymbol{F}_t$ is the external force exerted on the object (in this paper it is task-related). All the variables are expressed in the world coordinate system.

III. TASK-ORIENTED NULL-SPACE MANIPULABILITY OPTIMIZATION

To utilize each robot's maximum capability, its force polytope needs to be optimized with respect to the manipulation task. As WFME is a conservative approximation of the force polytope as shown in Fig. 3, we optimize WFME instead to maximize each robot's force capability. To do this, we first define the nominal task ellipsoid to encode the task's force characteristics. Second, we optimize the robot's null-space motion to match the WFME with the nominal task ellipsoid.

A. Nominal Task Ellipsoid

The nominal task ellipsoid is defined as an ellipsoid of revolution, also called a prolate spheroid, whose two principle axes have the same length, while the third principle axis is the longest. The nominal task ellipsoid can be generated by the transformation of unit sphere as

$$\{\boldsymbol{c}_e = \boldsymbol{C} \boldsymbol{c}_s \mid \|\boldsymbol{c}_s\| \leqslant 1\}, \qquad (5)$$

where $\boldsymbol{c}_e$ is the Cartesian coordinates of the nominal task ellipsoid, $\boldsymbol{c}_s$ is the Cartesian coordinates of the unit sphere. $\boldsymbol{C}$ is the transformation matrix and can be calculated as $\boldsymbol{C} = \boldsymbol{R}_r \boldsymbol{R}_s$ with $\boldsymbol{R}_s$ being the scaling matrix and $\boldsymbol{R}_r$ being the rotation matrix. The scaling matrix $\boldsymbol{R}_s$ is defined as $\boldsymbol{R}_s = \text{diag}(l_x, l_y, l_z)$, where l_x, l_y, l_z, are the lengths of the principle axes. In this paper, $l_x = 1$ corresponds to the longest
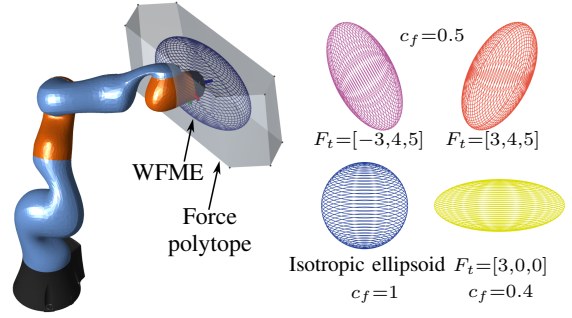

Fig. 3: Force polytope and WFME (left). Nominal task ellipsoids corresponding to different manipulation tasks where F_t represents the desired force along different directions (right).

axis that is aligned with the desired force. $l_y = l_z = c_f l_x$ correspond to the other two axes and $c_f \in (0, 1]$ can be set according to the task requirements, *e.g.* $c_f = 1$ (isotropic ellipsoid in Fig. 3) for manipulation tasks that require force to be equally distributed along different directions, while $c_f < 1$ (other cases in Fig. 3) for manipulation tasks that require force along a specific direction.

Given the desired force $\boldsymbol{F}_t$ on the object and the unit vector $\boldsymbol{a}_x$ of the longest axis, the common perpendicular vector $\boldsymbol{u}$ and the included angle ϕ between them, can be obtained by

$$\boldsymbol{u} = \frac{\boldsymbol{F}_t \times \boldsymbol{a}_x}{\|\boldsymbol{F}_t \times \boldsymbol{a}_x\|} \quad \text{and} \quad \phi = acos\left(\frac{\boldsymbol{F}_t \cdot \boldsymbol{a}_x}{\|\boldsymbol{F}_t\| \|\boldsymbol{a}_x\|}\right), \qquad (6)$$

respectively. Therefore, the rotation matrix $\boldsymbol{R}_r$—which is used to align the longest axis with the desired force—can be obtained according to the angle-axis representation as

$$\boldsymbol{R}_r = \boldsymbol{E}_3 c_\phi + (1 - c_\phi) \boldsymbol{u} \boldsymbol{u}^T + \boldsymbol{u}^\times s_\phi, \qquad (7)$$

where $s_\phi = \sin(\phi)$, $c_\phi = \cos(\phi)$, $\boldsymbol{E}_3$ is the 3×3 identity matrix and $\boldsymbol{u}^\times$ is the skew symmetric matrix of $\boldsymbol{u}$.

From (5), we can obtain the following equation

$$\boldsymbol{c}_e^T \boldsymbol{M}_t^{-1} \boldsymbol{c}_e \leqslant 1, \quad \text{with } \boldsymbol{M}_t = \boldsymbol{C} \boldsymbol{C}^T, \qquad (8)$$

where the symmetric positive definite matrix $\boldsymbol{M}_t$ represents the nominal task ellipsoid. A few nominal task ellipsoids for different manipulation tasks are visualized in Fig. 3. The shape of nominal task ellipsoid is determined by the coefficient c_f, while its orientation is determined by the direction of the desired force $\boldsymbol{F}_t$.

B. Null-Space Manipulability Optimization

As multiple robots manipulate the object jointly, the end-effector of each robot is assumed to be fixed on the object via the corresponding grasping point. Thus, the velocity of each robot's end-effector $\dot{\boldsymbol{x}}_k$ can be derived from the velocity of the object $\dot{\boldsymbol{x}}_o$ as

$$\dot{\boldsymbol{x}}_k = \begin{bmatrix} \boldsymbol{E}_3 & -\boldsymbol{r}_k^\times \\ \boldsymbol{0} & \boldsymbol{E}_3 \end{bmatrix} \dot{\boldsymbol{x}}_o = \boldsymbol{G}_{pk}^T \dot{\boldsymbol{x}}_o, \qquad (9)$$

where $\boldsymbol{G}_{pk}$ is the grasp matrix of robot k, $\boldsymbol{r}_k$ is the position vector from the object's center of mass to the grasping point of robot k, and $\dot{\boldsymbol{x}}_o = [\boldsymbol{v}_o^T, \boldsymbol{\omega}_o^T]^T$. By differentiating (9) the acceleration constraint can be obtained as

$$\ddot{\boldsymbol{x}}_k = \boldsymbol{G}_{pk}^T \ddot{\boldsymbol{x}}_o + \dot{\boldsymbol{G}}_{pk}^T \dot{\boldsymbol{x}}_o. \qquad (10)$$

At the same time, the pseudo-inverse solutions for joint velocity $\dot{\boldsymbol{\Theta}}_k$ and acceleration $\ddot{\boldsymbol{\Theta}}_k$ of each robot are

$$\dot{\boldsymbol{\Theta}}_k = \boldsymbol{J}_k^{\dagger} \dot{\boldsymbol{x}}_k \text{ and } \ddot{\boldsymbol{\Theta}}_k = \boldsymbol{J}_k^{\dagger} \left(\ddot{\boldsymbol{x}}_k - \dot{\boldsymbol{J}}_k \dot{\boldsymbol{\Theta}}_k \right), \quad (11)$$

respectively; where $\boldsymbol{J}_k^{\dagger}$ is the pseudo-inverse of $\boldsymbol{J}_k$.

In this paper, the null-space motion of the redundant manipulators is used to optimize the force manipulability. Given the current WFME of robot k and a desired nominal task ellipsoid (Section III-A), using the tensor representation and exploiting the fact that ellipsoids lie on the Riemannian manifold of symmetric positive definite (SPD) matrices [18], the velocity-level inverse kinematics with task-oriented manipulability optimization can be derived as

$$\dot{\boldsymbol{\Theta}}_k = \boldsymbol{J}_k^{\dagger} \dot{\boldsymbol{x}}_k + \left(\boldsymbol{I} - \boldsymbol{J}_k^{\dagger} \boldsymbol{J}_k \right) \left(\boldsymbol{\mathcal{J}}_{M_k}^{f\dagger} \right)^T \boldsymbol{K}_M \operatorname{vec} \left(\dot{\boldsymbol{M}}_k^f \right). \quad (12)$$

The first term of the right hand side is same as (11), while the second term projects the difference between the nominal task ellipsoid and WFME to the null space motion of the manipulator. $\boldsymbol{K}_M$ is the scaling matrix, $\operatorname{vec}()$ denotes the vectorization of symmetric matrices with Mandel notation[1], $\dot{\boldsymbol{M}}_k^f = \operatorname{Log}_{\boldsymbol{M}_k^f} \boldsymbol{M}_t$, the operator Log is logarithm map [25] which can find the tangent vector between two points in the SPD manifold. $\boldsymbol{M}_k^f$ represents the WFME of robot k and is defined as

$$\boldsymbol{M}_k^f = \left(\boldsymbol{J}_k \boldsymbol{W}_k^T \boldsymbol{W}_k \boldsymbol{J}_k^T \right)^{-1}. \quad (13)$$

The force manipulability Jacobian $\boldsymbol{\mathcal{J}}_{M_k}^f$ projects the scaled rate of change of $\boldsymbol{M}_k^f$ to the joint velocity $\dot{\boldsymbol{\Theta}}_k$ and it is obtained as

$$\begin{aligned} \boldsymbol{\mathcal{J}}_{M_k}^f &= -\frac{\partial \boldsymbol{M}_k^{f-1}}{\partial \boldsymbol{\Theta}_k} \times_1 \boldsymbol{M}_k^f \times_2 \boldsymbol{M}_k^f, \\ \frac{\partial \boldsymbol{M}_k^{f-1}}{\partial \boldsymbol{\Theta}_k} &= \frac{\partial \boldsymbol{J}_k}{\partial \boldsymbol{\Theta}_k} \times_2 \boldsymbol{J}_W + \frac{\partial \boldsymbol{J}_k^T}{\partial \boldsymbol{\Theta}_k} \times_1 \boldsymbol{J}_W, \end{aligned} \quad (14)$$

where $\boldsymbol{J}_W = \boldsymbol{J}_k \boldsymbol{W}_k^T \boldsymbol{W}_k$, $\times_n$ is the n-mode tensor product[2] [26]. The force capability optimization is achieved in the null space of manipulation task by using the proposed task-oriented manipulability optimization (12).

IV. DECENTRALIZED ABILITY-AWARE ADAPTIVE CONTROL

According to the force polytope shown in Fig. 3, the maximum operational force along any specific direction can be calculated. Subsequently, the decentralized ability-aware adaptive controller (DA^3C) computes the control inputs in accordance with the force capability of each robot.

A. Force Capability

The maximum operational force of each manipulator along the specific task direction $\boldsymbol{F}_t$ is calculated by

$$\max_{\boldsymbol{F}_k} \qquad \|\boldsymbol{F}_k\|_2 \qquad (15a)$$

$$\text{s.t.} \quad |\boldsymbol{H}_k \ddot{\boldsymbol{\Theta}}_k + \boldsymbol{C}_k + \boldsymbol{G}_k + \boldsymbol{J}_k^T \boldsymbol{F}_k| \leq \boldsymbol{\tau}_{k,max} \quad (15b)$$

$$\boldsymbol{F}_t^{\times} \boldsymbol{F}_k = \boldsymbol{0}, \qquad (15c)$$

---

[1]The Mandel representation $[\boldsymbol{\varepsilon}]$ (as a column-vector) of any second rank, symmetric tensor $\boldsymbol{\varepsilon}$ is defined as follows: $[\boldsymbol{\varepsilon}] = [\varepsilon_{11}, \varepsilon_{22}, \sqrt{2}\varepsilon_{12}]^T$.

[2]1-mode and 2-mode: $\boldsymbol{A} \times_1 \boldsymbol{U}^T = \boldsymbol{U}^T \boldsymbol{A}$, $\boldsymbol{A} \times_2 \boldsymbol{V}^T = \boldsymbol{A}\boldsymbol{V}$

where $\boldsymbol{H}_k$ is the inertia matrix of robot k, $\boldsymbol{C}_k$ is the Coriolis and centrifugal force of robot k, $\boldsymbol{G}_k$ is the gravity of robot k. The force polytope is defined by (15b) and can be obtained from the dynamic equation of the manipulator, while (15c) is used to define the specific task direction.

B. Ability-Aware Adaptive Controller

In order to track the desired trajectory of the object during decentralized multi-robot collaborative manipulation, we propose an ability-aware adaptive controller in which the force capability of each robot is considered.

According to (4), the ideal reference dynamics model of the object can be written as

$$\ddot{\boldsymbol{x}}_o^* = \boldsymbol{A}^* \dot{\boldsymbol{x}}_o^* + \boldsymbol{B}^* \left(\boldsymbol{F}_t^* - \boldsymbol{N}_{cg} \right), \quad (16)$$

where $\boldsymbol{A}^*$ is a Hurwitz stable matrix written as $\boldsymbol{A}^* = \begin{bmatrix} -\frac{\mu_l}{m_o^*} \boldsymbol{E}_3 & \boldsymbol{0} \\ \boldsymbol{0} & -\mu_r \boldsymbol{I}_o^{*-1} \end{bmatrix}$, $\boldsymbol{B}^* = \begin{bmatrix} \frac{1}{m_o^*} \boldsymbol{E}_3 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_o^{*-1} \end{bmatrix}$, and $\boldsymbol{N}_{cg} = \begin{bmatrix} -m_o^* \boldsymbol{g} \\ \boldsymbol{\omega}_o \times (\boldsymbol{I}_o^* \boldsymbol{\omega}_o) \end{bmatrix}$ is a stacked vector of gravity term and nonlinear term in (4), and m_o^* and $\boldsymbol{I}_o^*$ are nominal mass and inertia matrix of the object. Given the bounded reference (desired) trajectory $\dot{\boldsymbol{x}}_o^*$ and $\ddot{\boldsymbol{x}}_o^*$ of the object, the bounded reference control input $\boldsymbol{F}_t^*$, which represents the external force exerted on the object, can be computed from (16).

To design the adaptive controller, the actual object's dynamics model can be rewritten in the following linear form with respect to the system state $\dot{\boldsymbol{x}}_o$ as

$$\ddot{\boldsymbol{x}}_o = \boldsymbol{A} \dot{\boldsymbol{x}}_o + \sum_{k=1}^K \boldsymbol{B}_k \left(\boldsymbol{F}_k - \boldsymbol{U}_k \right), \quad (17)$$

where $\boldsymbol{A}$ and $\boldsymbol{B}_k$ are unknown constant matrices, $\boldsymbol{A} = -\begin{bmatrix} \frac{\mu_l}{m_o} \boldsymbol{E}_3 & \boldsymbol{0} \\ \boldsymbol{0} & \mu_r \boldsymbol{I}_o^{-1} \end{bmatrix}$, $\boldsymbol{B}_k = \begin{bmatrix} \frac{1}{m_o} \boldsymbol{E}_3 & \boldsymbol{0} \\ \boldsymbol{I}_o^{-1} \boldsymbol{r}_k^{\times} & \boldsymbol{I}_o^{-1} \end{bmatrix}$, $\boldsymbol{F}_k$ is the input of robot k, K is the total number of robots, $\boldsymbol{U}_k = \boldsymbol{W}_{\phi k}^{*T} \boldsymbol{\Phi}_k$ is unknow nonlinear function caused by modelling uncertainties which can be approximated by a Radial Basis Functions (RBFs) neural network, $\boldsymbol{W}_{\phi k}^*$ is the weight matrix for the RBFs, $\boldsymbol{\Phi}_k$ is the vector of RBFs and output bias.

For each robot, the adaptive control input is designed as

$$\boldsymbol{F}_k = \boldsymbol{K}_{xk}^T \dot{\boldsymbol{x}}_o + \boldsymbol{K}_{rk}^T \boldsymbol{F}_t^* + \boldsymbol{K}_{nk}^T \boldsymbol{N}_{cg} + \boldsymbol{W}_{\phi k}^T \boldsymbol{\Phi}_k, \quad (18)$$

where $\boldsymbol{K}_{xk}$, $\boldsymbol{K}_{rk}$, $\boldsymbol{K}_{nk}$ and $\boldsymbol{W}_{\phi k}$ are control gain matrices. Considering the force capability of each robot, the control input constraints of DA^3C are guaranteed by positive μ-modification [27]. Therefore, the modified ideal reference dynamics model (see (16)) and the actual system dynamics (see (17)) with input constraints are rewritten as

$$\ddot{\boldsymbol{x}}_o^* = \boldsymbol{A}^* \dot{\boldsymbol{x}}_o^* + \boldsymbol{B}^* \left(\boldsymbol{F}_t^* + \sum_{k=1}^K \boldsymbol{K}_{fk}^T \Delta \boldsymbol{F}_k - \boldsymbol{N}_{cg} \right), \quad (19)$$

$$\ddot{\boldsymbol{x}}_o = \boldsymbol{A} \dot{\boldsymbol{x}}_o + \sum_{k=1}^K \boldsymbol{B}_k \left(\boldsymbol{F}_k + \Delta \boldsymbol{F}_k - \boldsymbol{U}_k \right), \quad (20)$$

where $\boldsymbol{K}_{fk}$ is the control gain matrix, $\Delta \boldsymbol{F}_k$ is the control deficiency, which is described in Appendix A, and is used to generate the adaptive augmentation for the reference model.
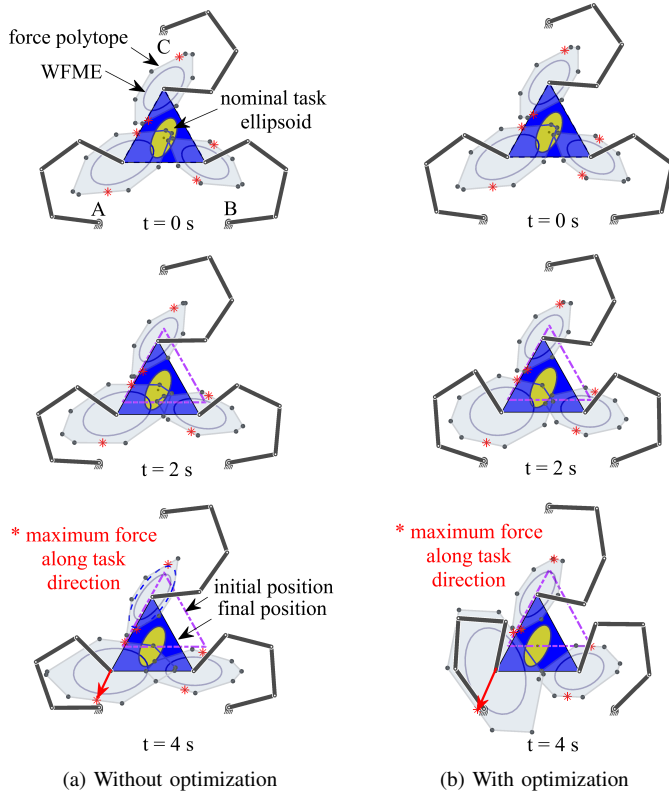
Fig. 4: Simulation results with and without manipulability optimization.

Furthermore, by defining the tracking error $e = \dot{x}_o - \dot{x}_o^*$, the tracking error dynamics can be obtained as

$$\dot{e} = \ddot{x}_o - \ddot{x}_o^* = A^* e - \sum_{k=1}^{K} B^* \tilde{K}_{fk}^T \Delta F_k +$$
$$\sum_{k=1}^{K} B_k \left(\tilde{K}_{xk}^T \dot{x}_o + \tilde{K}_{rk}^T F_t^* + \tilde{K}_{nk}^T N_{cg} + \tilde{W}_{\phi k}^T \Phi_k \right), \quad (21)$$

where $\tilde{K} = K - K^*$ is the error matrix of control gain and K^* is ideal control gain (see Appendix B).

Last, the following adaptive laws are adopted in order to guarantee asymptotic tracking of the reference trajectory.

$$\dot{\tilde{K}}_{xk} = -\Gamma_{xk} \dot{x}_o e^T P B_k, \quad \dot{\tilde{K}}_{rk} = -\Gamma_{rk} F_t^* e^T P B_k,$$
$$\dot{\tilde{K}}_{nk} = -\Gamma_{nk} N_{cg} e^T P B_k, \quad \dot{\tilde{W}}_{\phi k} = -\Gamma_{\phi k} \Phi e^T P B_k, \quad (22)$$
$$\dot{\tilde{K}}_{fk} = \Gamma_{fk} \Delta F_k e^T P B^*,$$

where P is unique SPD solution of the Lyapunov equation, $P A^* + A^{*T} P = -Q < 0$, where Q is any SPD matrix; Γ_{xk}, Γ_{rk}, Γ_{nk}, $\Gamma_{\phi k}$, and Γ_{fk} are SPD gain matrices. The proof for asymptotic stability and asymptotic tracking using the above adaptive laws can be found in Appendix B.

C. Computed-Torque Control for Each Robot

According to the constrained operational force and the adaptive reference trajectory of the object, the controller of each robot can be designed as the following well known computed-torque feedforward control,

$$\tau_k = H_k \left(\ddot{\Theta}_k + K_p \tilde{\Theta}_k + K_v \dot{\tilde{\Theta}}_k \right)$$
$$+ C_k + G_k + J_k^T \left(F_k + \Delta F_k \right), \quad (23)$$

where $\tilde{\Theta}_k$, $\dot{\tilde{\Theta}}_k$ are the error vectors of joint angle and joint angular velocity, and K_p, K_v are the corresponding gains.

D. Level of Communication

In the proposed ability-aware adaptive controller, the levels of communication between the robots in the team may vary depending on the different states of the multi-robot system. As long as the desired operational force of each robot is within its force polytope, each robot can be controlled in a fully decentralized manner. Each robot should know the object's velocity and the grasping location on the object. However, in cases where a robot would need to exceed its capabilities to track the reference input—desired operational force lies outside of the force polytope—then, the control deficiency ΔF_k of each robot should be broadcast to all the robots of the team. The control deficiency of all the robots will be used in (19) to modify the unfeasible reference control input for each robot, which results in a coordinated adaptation of the robot team.

V. RESULTS

First, we perform three ablation studies to demonstrate the benefits of each one of the proposed components and their combination. Second, we demonstrate the adaptation capabilities of the DA^3C framework in a decentralized manipulation setup. Third, we validate DA^3C on a physical simulation, where three heterogeneous robots manipulate an object. The control frequency of all the simulations is set to 500 Hz.

A. Ablation Studies

1) Force Capabilities with and without Manipulability Optimization: In this ablation study, three 4-DOF planar robots manipulate an object from initial position [0, 0] m to final position [-0.1, -0.2] m. The maximum joint torques of robot A, B and C are set to 0.8, 0.6 and 0.6 N m, respectively. The simulation results without and with task-oriented manipulability optimization are shown in Fig. 4. Using the proposed method the WFME of each robot is optimized through the null-space motion to track the nominal task ellipsoid (yellow ellipsoid), and consequently the force capability (red arrow) along the specific task direction is increased.

2) Ability-Aware vs Ability-Agnostic Adaptive Controller: The second ablation study compares the tracking performance of the proposed ability-aware adaptive controller (DA^3C) with an ability-agnostic adaptive controller [1]. The task considers three robots manipulating an object, with $20\,\text{kg}$ mass and $20\,\text{kg m}^2$ inertia (along z-axis), on a plane with a sliding friction coefficient of 0.2. The adaptive controllers are initiated with 80% of these values. The reference control input is set to $F_t^* = 4 * [\sin(0.4t), \sin(0.3t)]^T$ N. The maximum force F_{max} of each robot is set to [1.0, 1.0] N, and the constant vector δ (see Appendix A) is set to 10% of F_{max}. The object's trajectories with ability-agnostic control (green line) and ability-aware control (blue line) are shown in Fig. 5. For the ability-agnostic adaptive controller, the control input (see Fig. 6) exceed the force constraints, hence, we limit
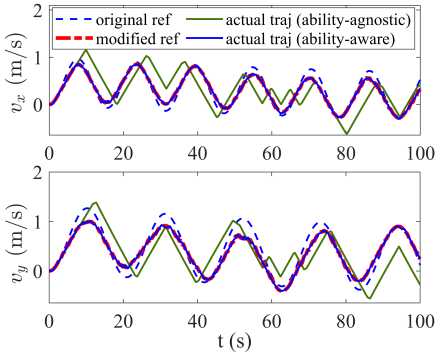
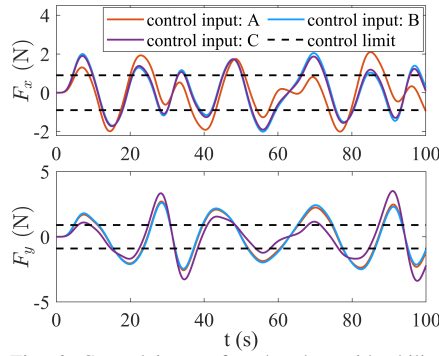Fig. 5: State of the object with ability-agnostic and ability-aware controller.

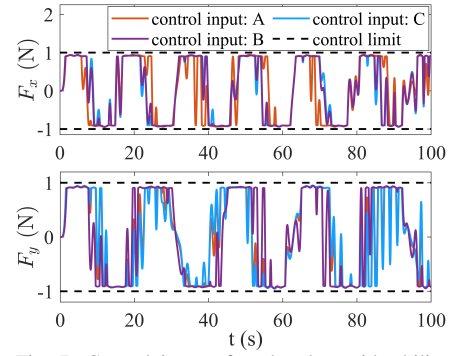Fig. 6: Control input of each robot with ability-agnostic controller.

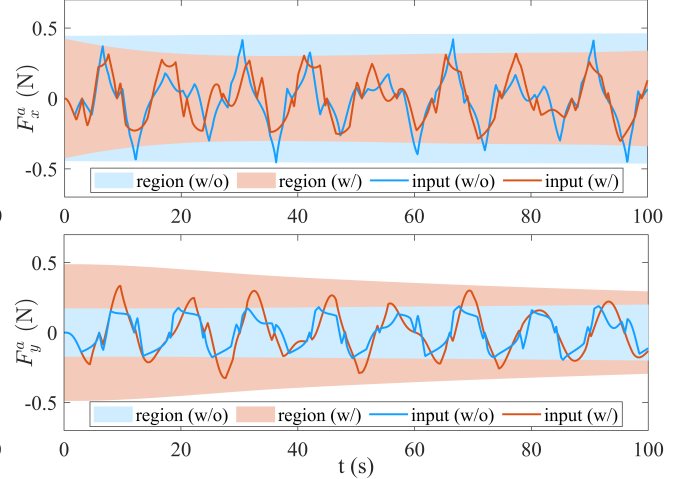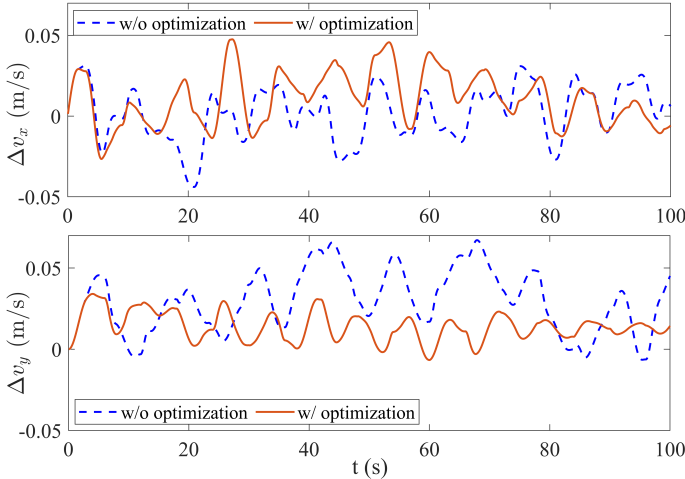Fig. 7: Control input of each robot with ability-aware controller.

Fig. 8: Tracking errors of the object (1st column) and control inputs of robot A (2nd column) with and without manipulability optimization. The cyan regions represent the force capability envelope of the robot without optimization while the orange regions represents the capability envelope with optimization.
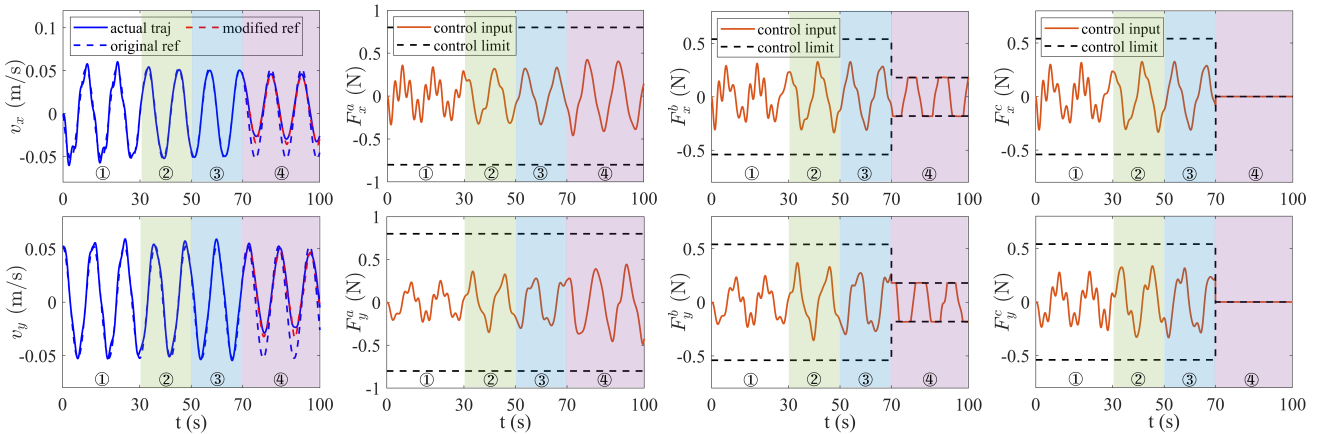

Fig. 9: State of the object (1st column) and control inputs of robot A (2nd column) and robot B (3rd column) and robot C (4th column). In region ②, the mass of the object is increased from 20 kg to 30 kg. In region ③, the friction coefficient is decreased from 0.2 to 0.1, while in region ④, the force capability of robot B is reduced to 1/3 while robot C is shut off.

the control input to the maximum force. This results in a significant deviation of the object's trajectory from the desired one (see Fig. 5). On the other hand, given the limited capability of each robot in the ability-aware controller (see Fig. 7) the new reference trajectory (red dashed line) deviates from the original one (blue dashed line) according to the control deficiency of all the robots as shown in Fig. 5, such that the multi-robot system can track the modified trajectory accurately with average tracking error 0.009 m/s and 0.013 m/s along x-axis and y-axis, respectively.

3) DA³C with and without Manipulability Optimization: Here, we compare the proposed ability-aware adaptive control with and without manipulability optimization. The three manipulators of Section V-A1) are used to manipulate the same object as in Section V-A2). The reference trajectory is circular and is described with $v_x = -0.1 * \frac{\pi}{6} * \sin(\frac{\pi}{6}t)$ m/s, $v_y = 0.1 * \frac{\pi}{6} * \cos(\frac{\pi}{6}t)$ m/s and $w_z = 0$. The nominal task ellipsoid is a sphere to equally allot the force capabilities along different directions. The velocity tracking errors of the object are shown in Fig. 8 (left). The average tracking errors along two axes with manipulability optimization are $(0.0138, 0.0143)$

m/s, which outperforms (0.0138, 0.0301) m/s corresponding to the one without. In the right part of Fig. 8, we can observe all the control inputs along with the force limits (regions). The DA^3C with manipulability optimization regulates the robot's configuration to achieve similar force capabilities along both directions, such that tracking performance is balanced in both directions.

B. DA^3C on-the-fly Adaptation

Here, we show that the proposed method is capable of handling changes with respect to the mass of the object, the friction coefficient and also tolerant to faults of team members. We consider the same circular task, where the mass of the object is increased from 20 kg (white region ①) to 30 kg (green region ②) at 30 s, the friction coefficient is decreased from 0.2 (green region ②) to 0.1 (blue region ③) at 50 s, and the max force of robot B is reduced to 1/3 while robot C is shut off (purple region ④) at 70 s. The results are shown in Fig. 9, the control inputs of the robots increase when the object's mass is increased and decrease when the friction coefficient is decreased. Further, it is worth noting that in the purple region ④, the control input of robot A is increased and the reference trajectory is adapted, due to the loss of robot C and the reduced capability of robot B.

C. DA^3C for Multi-Robot Collaboration

To verify the proposed DA^3C framework, three heterogeneous robots, the torque-controlled Kuka-iiwa (A), the Franka-panda (B), and one position-based admittance controlled mobile manipulator (C), are used to manipulate an object on a surface (see in Fig. 10), where both the object's mass, and the friction between the object and the surface are unknown. The task is a circular motion and the maximum force for all the robots is limited to 10 N, while the constant $\boldsymbol{\delta}$ (see Appendix A) are set to 1, 2, and 3 N, respectively. For the torque-controlled robots, the feedback gains $\boldsymbol{K}_p$, $\boldsymbol{K}_v$ in (23) are set to diag([16, 16, 16, 12, 12, 12, 12]) and diag([0.2, 0.2, 0.2, 0.1, 0.1, 0.1, 0.1]), respectively. The interaction between the object and the robots is modelled with a spring-damper (stiffness 5000 N/m and damping 500 Ns/m). The physical simulation and visualization are developed in MATLAB Simscape and Simulink.

The object's desired and actual trajectories, and the force profiles of each robot are shown in Fig. 11. The reference trajectory is modified according to the control deficiency, due to the limits on the robots' force capabilities. Robot C does not always track the desired force accurately, due to the admittance controller. Thus, its safety zone is set larger (green area) to always experience forces within its limits. Also, such force deviations introduce further unmodelled interaction forces, due to the formed closed-chain. Yet, each robot adapts its control gains on-the-fly to cope with these deviations too.

VI. CONCLUSION

In this paper we propose a decentralized ability-aware adaptive control (DA^3C) framework for multi-robot collaborative

manipulation, which can handle uncertain system parameters, input constraints and band-limited communication. The key idea is that the force capability of each robot is maximized by exploiting its null-space motion, while the designed adaptive controller enables decentralized coordination according to the capability of each robot. The proposed method achieves accurate trajectory tracking irrespective of the low-level controllers, and can be used for heterogeneous fixed-base and mobile-base multi-robot systems. An open challenge is the inclusion of joint position limits into the ability-aware adaptive controller. In our future work, we plan to use DA^3C for human-robot co-manipulation experiments where the access to human's capability is not straightforward.

APPENDIX A
CONTROL INPUT WITH μ-MODIFICATION

The control input with μ-modification is written as

$$\bar{\boldsymbol{F}}_k = \frac{1}{1+\mu}\left(\boldsymbol{F}_k + \mu \boldsymbol{F}_{kmax}^\delta \mathrm{sat}\left(\frac{\boldsymbol{F}_k}{\boldsymbol{F}_{kmax}^\delta}\right)\right)$$

$$= \begin{cases} \boldsymbol{F}_k, & |\boldsymbol{F}_k| \leqslant \boldsymbol{F}_{kmax}^\delta \\ \frac{1}{1+\mu}\left(\boldsymbol{F}_k + \mu \boldsymbol{F}_{kmax}^\delta\right), & \boldsymbol{F}_k > \boldsymbol{F}_{kmax}^\delta \\ \frac{1}{1+\mu}\left(\boldsymbol{F}_k - \mu \boldsymbol{F}_{kmax}^\delta\right), & \boldsymbol{F}_k < -\boldsymbol{F}_{kmax}^\delta \end{cases} \quad (24)$$

where μ is positive design constant, $\boldsymbol{F}_{kmax}^\delta = \boldsymbol{F}_{kmax} - \boldsymbol{\delta}$, $\boldsymbol{\delta}$ is a constant vector, $\boldsymbol{0} < \boldsymbol{\delta} < \boldsymbol{F}_{kmax}$, and $\boldsymbol{F}_{kmax}$ is the maximum force of robot k which is obtained from (15). The corresponding control deficiency can be calculated as

$$\Delta \boldsymbol{F}_k = \boldsymbol{F}_{kmax}\mathrm{sat}\left(\frac{\bar{\boldsymbol{F}}_k}{\boldsymbol{F}_{kmax}}\right) - \boldsymbol{F}_k. \quad (25)$$

APPENDIX B
LYAPUNOV STABILITY ANALYSIS

In order to match (19) and (20), we can choose the ideal gain matrices $\boldsymbol{K}_{xk}^*$, $\boldsymbol{K}_{rk}^*$, $\boldsymbol{K}_{fk}^*$, and $\boldsymbol{K}_{nk}^*$ according to the following forms:

$$\boldsymbol{A} + \sum_{k=1}^K \boldsymbol{B}_k \boldsymbol{K}_{xk}^{*T} = \boldsymbol{A}^*, \quad \sum_{k=1}^K \boldsymbol{B}_k \boldsymbol{K}_{rk}^{*T} = \boldsymbol{B}^*,$$

$$-\sum_{k=1}^K \boldsymbol{B}_k \boldsymbol{K}_{nk}^{*T} = \boldsymbol{B}^*, \quad \boldsymbol{B}^* \boldsymbol{K}_{fk}^{*T} = \boldsymbol{B}_k. \quad (26)$$

According to the error dynamics in (21), we consider the following Lyapunov function candidate:

$$V\left(\boldsymbol{e}, \tilde{\boldsymbol{K}}\right) = \boldsymbol{e}^T \boldsymbol{P} \boldsymbol{e} + \sum_{k=1}^K \mathrm{tr}\left(\tilde{\boldsymbol{K}}_{xk}^T \boldsymbol{\Gamma}_{xk}^{-1} \tilde{\boldsymbol{K}}_{xk}\right)$$

$$+ \sum_{k=1}^K \mathrm{tr}\left(\tilde{\boldsymbol{K}}_{rk}^T \boldsymbol{\Gamma}_{rk}^{-1} \tilde{\boldsymbol{K}}_{rk}\right) + \sum_{k=1}^K \mathrm{tr}\left(\tilde{\boldsymbol{K}}_{nk}^T \boldsymbol{\Gamma}_{nk}^{-1} \tilde{\boldsymbol{K}}_{nk}\right) \quad (27)$$

$$+ \sum_{k=1}^K \mathrm{tr}\left(\tilde{\boldsymbol{W}}_{\phi k}^T \boldsymbol{\Gamma}_{\phi k}^{-1} \tilde{\boldsymbol{W}}_{\phi k}\right) + \sum_{k=1}^K \mathrm{tr}\left(\tilde{\boldsymbol{K}}_{fk}^T \boldsymbol{\Gamma}_{fk}^{-1} \tilde{\boldsymbol{K}}_{fk}\right),$$

By using the adaptive law (22), the derivative of Lyapunov function[3] decreases along the tracking error dynamics as

$$\dot{V}\left(\boldsymbol{e}, \tilde{\boldsymbol{K}}_{xk}, \tilde{\boldsymbol{K}}_{rk}, \tilde{\boldsymbol{K}}_{nk}, \tilde{\boldsymbol{K}}_{fk}, \tilde{\boldsymbol{W}}_{\phi k}\right) = -\boldsymbol{e}^T \boldsymbol{Q} \boldsymbol{e} \leqslant 0. \quad (28)$$

---

[3]For two real column matrices $\boldsymbol{a} \in \mathbb{R}^n$ and $\boldsymbol{b} \in \mathbb{R}^n$: $\mathrm{tr}\left(\boldsymbol{b}\boldsymbol{a}^T\right) = \boldsymbol{a}^T \boldsymbol{b}$.
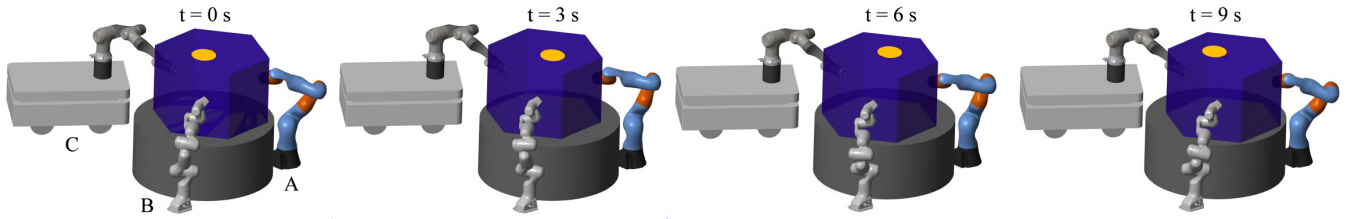
Fig. 10: A heterogeneous multi-robot team manipulates an object. The object's circular motion is illustrated with respect to the yellow circle, which is fixed in the world frame.
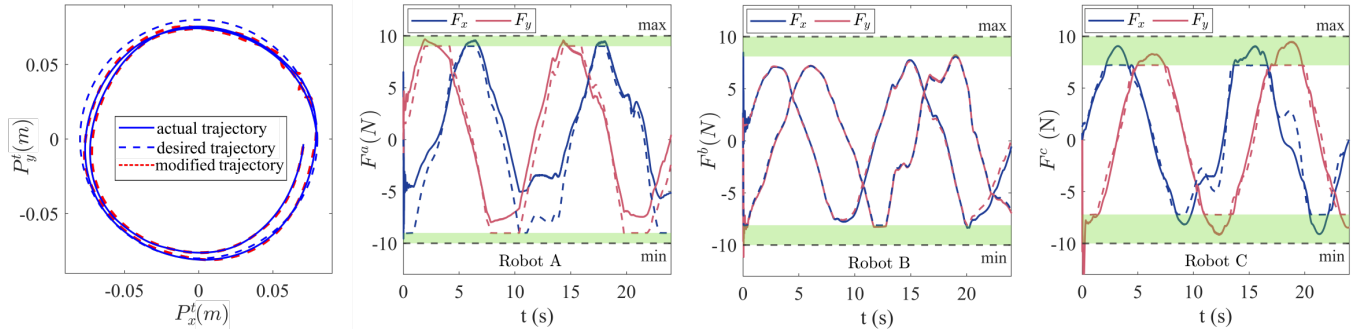

Fig. 11: Desired, modified reference and actual trajectory of the object (1st column), and desired (dashed line) and actual (solid line) end-effector forces of robot A (2nd column), robot B (3rd column) and robot C (4th column). By setting robot-specific safety zones (green region), the force of each robot is guaranteed not to violate its actual limits after a few iterations.

Therefore, given a bounded reference input, we can conclude that the system can achieve asymptotic tracking by using Barbalat's Lemma.

REFERENCES

[1] P. Culbertson and M. Schwager, "Decentralized adaptive control for collaborative manipulation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 278–285.

[2] T. Stouraitis, I. Chatzinikolaidis, M. Gienger, and S. Vijayakumar, "Online hybrid motion planning for dyadic collaborative manipulation via bilevel optimization," *IEEE Transactions on Robotics*, vol. 36, no. 5, pp. 1452–1471, 2020.

[3] Y. Emam, S. Mayya, G. Notomista, A. Bohannon, and M. Egerstedt, "Adaptive task allocation for heterogeneous multi-robot teams with evolving and unknown robot capabilities," *arXiv preprint arXiv:2003.03344*, 2020.

[4] T. Yoshikawa, "Manipulability of robotic mechanisms," *The International Journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.

[5] ——, "Manipulability and redundancy control of robotic mechanisms," in *IEEE International Conference on Robotics and Automation*, 1985, pp. 1004–1009.

[6] S. Lee, "Dual redundant arm configuration optimization with task-oriented dual arm manipulability," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 1, pp. 78–97, 1989.

[7] G. Huber and D. Wollherr, "Efficient closed-form task space manipulability for a 7-dof serial robot," *Robotics*, vol. 8, no. 4, p. 98, 2019.

[8] P. Chiacchio, S. Chiaverini, L. Sciavicco, B. Siciliano *et al.*, "Global task space manipulability ellipsoids for multiple-arm systems," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 5, pp. 678–685, 1991.

[9] M. Azad, J. Babič, and M. Mistry, "Effects of the weighting matrix on dynamic manipulability of robots," *Autonomous Robots*, vol. 43, no. 7, pp. 1867–1879, 2019.

[10] P. Chiacchio, Y. Bouffard-Vercelli, and F. Pierrot, "Force polytope and force ellipsoid for redundant manipulators," *Journal of Robotic Systems*, vol. 14, no. 8, pp. 613–620, 1997.

[11] L. Yan, Z. Mu, W. Xu, and B. Yang, "Coordinated compliance control of dual-arm robot for payload manipulation: Master-slave and shared force control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 2697–2702.

[12] L. Yan, Y. Yang, W. Xu, and S. Vijayakumar, "Dual-arm coordinated motion planning and compliance control for capturing moving objects with large momentum," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 7137–7144.

[13] L. Jin, S. Li, H. M. La, and X. Luo, "Manipulability optimization of redundant manipulators using dynamic neural networks," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 4710–4720, 2017.

[14] A. Ajoudani, N. G. Tsagarakis, and A. Bicchi, "Choosing poses for force and stiffness control," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1483–1490, 2017.

[15] D. Busson, R. Bearee, and A. Olabi, "Task-oriented rigidity optimization for 7 dof redundant manipulators," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 14 588–14 593, 2017.

[16] L. Yan, W. Xu, Z. Hu, and B. Liang, "Multi-objective configuration optimization for coordinated capture of dual-arm space robot," *Acta Astronautica*, vol. 167, pp. 189–200, 2020.

[17] L. Rozo, N. Jaquier, S. Calinon, and D. G. Caldwell, "Learning manipulability ellipsoids for task compatibility in robot manipulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 3183–3189.

[18] N. Jaquier, L. D. Rozo, D. G. Caldwell, and S. Calinon, "Geometry-aware tracking of manipulability ellipsoids." in *Robotics: Science and Systems*, 2018, pp. 1–9.

[19] Y.-H. Liu and S. Arimoto, "Decentralized adaptive and nonadaptive position/force controllers for redundant manipulators in cooperations," *The International Journal of Robotics Research*, vol. 17, no. 3, pp. 232–247, 1998.

[20] F. Aghili, "Adaptive control of manipulators forming closed kinematic chain with inaccurate kinematic model," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 5, pp. 1544–1554, 2012.

[21] G. Joshi and G. Chowdhary, "Deep model reference adaptive control," in *IEEE Conference on Decision and Control*, 2019, pp. 4601–4608.

[22] G.-B. Dai and Y.-C. Liu, "Distributed coordination and cooperation control for networked mobile manipulators," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 5065–5074, 2016.

[23] Y. Ren, S. Sosnowski, and S. Hirche, "Fully distributed cooperation for networked uncertain mobile manipulators," *IEEE Transactions on Robotics*, 2020.

[24] P. B. G. Dohmann and S. Hirche, "Distributed control for cooperative manipulation with event-triggered communication," *IEEE Transactions on Robotics*, 2020.

[25] X. Pennec, P. Fillard, and N. Ayache, "A riemannian framework for tensor computing," *International Journal of computer vision*, vol. 66, no. 1, pp. 41–66, 2006.

[26] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[27] E. Lavretsky and N. Hovakimyan, "Positive mu-modification for stable adaptation in the presence of input constraints," in *American Control Conference*, 2004, pp. 2545–2550.