Decentralized Adaptive Control for Collaborative Manipulation of Rigid Bodies

Preston Culbertson, Jean-Jacques E. Slotine, and Mac Schwager

***Abstract*—In this work, we consider a group of robots working together to manipulate a rigid object to track a desired trajectory in SE(3). The robots have no explicit communication network among them, and they do not know the mass or friction properties of the object, or where they are attached to the object. However we assume they share data from a common IMU placed arbitrarily on the object. To solve this problem, we propose a decentralized adaptive control scheme wherein each agent maintains and adapts its own estimate of the object parameters in order to track a reference trajectory. We present an analysis of the controller's behavior, and show that all closed-loop signals remain bounded, and that the system trajectory will almost always (except for initial conditions on a set of measure zero) converge to the desired trajectory. We study the proposed controller's performance using numerical simulations of a manipulation task in 3D, as well as hardware experiments which demonstrate our algorithm on a planar manipulation task. These studies, taken together, demonstrate the effectiveness of the proposed controller even in the presence of numerous unmodeled effects, such as discretization errors and complex frictional interactions.**

I. Introduction

As robots are fielded in applications ranging from disaster relief to autonomous construction, it is likely that such applications will, at times, require manipulation of objects which exceed a single robot's actuation capacities (e.g. debris removal, material transport) [1]. Notably, when humans undertake such tasks, they are able to work together to flexibly and reliably transport objects which are too massive for one person alone. We are interested in enabling similar capabilities in robot teams.

In this work, we consider the problem of cooperative manipulation, where a group of robots works together to manipulate a common payload. This problem has long been of significant interest to both the multi-robot systems community, and the broader robotics community [2], since it requires close collaboration between robots in addition to the numerous sensing and actuation challenges involved in any manipulation task.

While collaborative manipulation has been studied in literature [3]–[5], manipulation teams are still quite limited in their functionality. In part, this is due to their reliance on extensive prior information about the payload to be manipulated, and their need for high-bandwidth, two-way communication either with other agents or a centralized planner. The requirement of exact payload knowledge (i.e. the payload's mass, inertial properties, and the location of its center of mass) imposes the requirement that every payload be thoroughly calibrated prior to manipulation, in addition to making manipulation teams brittle in the face of parameter uncertainty. Further, the utilization of communication networks introduces not only an additional point of failure, but also latency, which can limit the team's speed and robustness.

In this work, we take a minimalist approach to collaborative manipulation, and present an adaptive controller for collaborative manipulation that requires neither prior payload characterization nor point-to-point communication between agents. To achieve this, we use an $SE(3)$ sliding model control architecture, together with a decentralized adaptation law, which allows the manipulation team to adapt to unknown payload dynamics without explicit communication; we also prove our controller's convergence to tracking a reference trajectory using Barbalat's lemma. We only require that the agents obtain information from a common IMU sensor located somewhere on the body, or on one of the manipulation robots. This IMU could be wired to all the robots, or may broadcast its measurements to all robots over a simple one-way broadcast network.

Intuitively, during manipulation each agent executes the proposed controller, with its individual set of parameters (or, equivalently, control gains). The agents then use measurements from the common IMU to adapt their local parameters, such that the location of the IMU asymptotically tracks a desired trajectory. Such an approach leads to a manipulation algorithm which can easily scale in the number of agents, and is flexible enough to manipulate a large variety of payloads, while tolerating both parameter uncertainty and single-agent failure.

This paper proposes control and adaptation laws for manipulation tasks in both the plane and three-dimensional space, in addition to a proof of convergence to the desired trajectory, as well as the boundedness of all closed-loop signals. We further validate the controller's performance both numerically, simulating a manipulation task in 3D, and experimentally, using a team of ground robots performing a planar manipulation task.

II. Prior Work

Collaborative manipulation has been a subject of both sustained and broad interest in multi-robot systems. Early work such as [6] first proposed multi-arm manipulation strategies which used a central computer to control each arm. This early work was followed with a flurry of interest, including [3], [7], where the authors study the dynamics and force allocation of redundant manipulators, and [8], which investigates various control strategies for multiple arms. A number of authors [9]–[11] have used impedance controllers for collaborative manipulation, in order to yield compliant object behavior, as well as to avoid large internal forces on the payload. An approach for characterizing such forces is proposed in [12].

While early work focused almost exclusively on multi-arm manipulation, often with centralized architectures and only a few robots, other authors have focused instead on collaborative manipulation with large teams of mobile robots. In [5] and [13], the authors move furniture with small mobile robots, with a focus on reducing the communication and sensing requirements of manipulation algorithms. Other works have studied various decentralized manipulation strategies, including potential fields [14], caging [4], and distributed control allocation via matrix pseduo-inverse [15].

The problem of collaborative manipulation with aerial vehicles (such as quadrotors) has received extensive recent interest. In [16], the authors consider the problem of collaborative lifting using quadrotors which rigidly grasp a common payload; they use a least-squares solution to allocate control efforts between agents. Further, the problem of aerial towing (manipulation via cables) has been studied in works such as [17]–[19]. The authors in [20] use a distributed optimization algorithm to perform collaborative manipulation of a cable-suspended load.

While collaborative manipulation has been studied in a variety of settings, most work often relies on restrictive assumptions, such as communication between agents or prior knowledge of the object parameters, which limit the practical applicability of their approaches. More recent work has focused on removing these assumptions. In [21], the authors propose a controller which allows a group of agents to reach a force consensus without communication, using only the payload dynamics; a leader can steer this consensus to have the object track a desired trajectory. Further, [22] proposes a communication-free controller for collaborative manipulation with aerial robots, using online optimization to allocate control effort. Other works, such as [23], [24] propose distributed schemes for estimating object parameters, and using these estimates for control, using communication between agents to reach consensus on estimated parameters.

Previous work has proposed applying adaptive control to collaborative manipulation. The authors of [25] propose a centralized adaptive control scheme, which they demonstrate experimentally with a pair of robot arms. In [26], the authors propose a centralized robust adaptive controller for a group of mobile manipulators performing a collaborative manipulation task.

Perhaps closest to this work is [27], [28], which also propose distributed adaptive control schemes for collaborative manipulation. This work differs from previous literature in adapting both inertial *and* geometric parameters of the body; previous methods require exact measurement of manipulators' relative positions from the center of mass in order to invert the grasp matrix [29], which relates wrenches applied at contact points to wrenches applied at the center of mass. We argue that this assumption is quite restrictive, since, in practice, this amounts to knowing exactly how the body's translational and rotational dynamics are coupled. Accurately localizing large numbers of manipulators (especially with respect to the center of mass) can be both time consuming and difficult; it is also unclear why the body's normalized first moment of mass (i.e. the center of mass) is assumed known, while its zeroth (mass) and second moments (inertia tensor) are assumed unknown. In contrast, our method can treat a much broader class of model parameters (including inertial, geometric, and other effects such as gravity and friction) as unknowns, and handle them through online adaptation. We believe that this provides greater robustness to uncertainty in these parameters, and eliminates the need to estimate these quantities before manipulation.

An early version of this work appeared in [30]; this work differs greatly from the original version. The most fundamental difference is that this work builds on the Slotine-Li adaptive controller [31], which allows for much more efficient parameterization than the general model reference adaptive controller in the original version. We do not constrain the measurement point to lie at the center of mass, and further introduce a sliding variable that allows for trajectory tracking instead of simple velocity control. Finally, we present new simulation and experimental results studying the novel controller.

Our basic algorithm can also be compared to quorum sensing [32], [33], a natural phenomenon in which organisms or cells can exchange information [34] and synchronize themselves using a noisy, shared environment, instead of using direct information exchange. Such mechanisms have been shown to reduce the influence of random noise on synchronization phenomena [32], allowing for meaningful coordination without centralized control or explicit communication. The proposed algorithm can be seen as analogous to quorum sensing, wherein the common object acts as the shared envi-

ronment, which allows the agents to coordinate their control actions to achieve tracking without using direct communication.

III. Problem Statement

We consider a rigid body which undergoes general motion (i.e. translation and rotation), during collaborative manipulation by a team of N autonomous agents. There exists a world-fixed (Newtonian) frame $\mathcal{F}$, as well as a body-fixed frame $\mathcal{B}$, which is centered at $\mathbf{b}_{cm}$, the body's center of mass. There also exists P, a general point fixed to the body, which is located at $\mathbf{r}_p$ in $\mathcal{B}$. The body has mass m and body-frame inertia of $\mathbf{I}_{cm}$ about its center of mass. Figure 1 shows a schematic of the system.

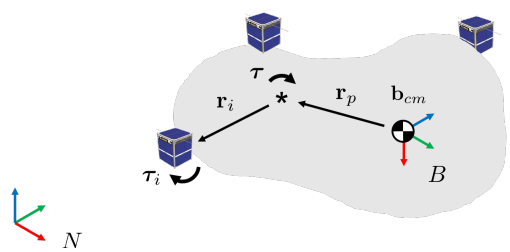


Fig. 1. Schematic of Collaborative Manipulation Task

Each agent is rigidly attached to the body at $\mathbf{r}_i$ from P, and applies a wrench $\boldsymbol{\tau}_i$, i.e. a combined force and torque. We assume the agents are attached rigidly to the object, and can apply a full wrench (both force and torque) at their attachment point.

We describe the body's pose in $\mathcal{F}$ using a configuration variable $\mathbf{q} = (\mathbf{x}, \mathbf{R}) \in \mathbb{R}^n \times SO(n)$, where

$$SO(n) = \left\{ \mathbf{R} \in \mathbb{R}^{n \times n} \mid \mathbf{R}^T \mathbf{R} = \mathbf{I}, \det(\mathbf{R}) = 1 \right\},$$

is the group of rotation matrices, $\mathbf{x}$ is the position of the reference point P in $\mathcal{F}$, $\mathbf{R}$ is the body-to-world rotation matrix relating $\mathcal{B}$ and $\mathcal{F}$, and $n \in \{2, 3\}$, depending on if the manipulation task is in the plane or in general 3D space, respectively. For simplicity, we say

$$\mathbb{R}^n \times SO(n) \cong SE(n),$$

i.e. we denote this set as the Special Euclidean group,

$$SE(n) = \left\{ \begin{bmatrix} \mathbf{R} & \mathbf{v} \\ \mathbf{0} & 1 \end{bmatrix} \mid \mathbf{R} \in SO(n), \mathbf{v} \in \mathbb{R}^n \right\}$$

which is the set of $(n+1) \times (n+1)$ transformation matrices, since these spaces are homeomorphic. Abusing notation, we define $\dot{\mathbf{q}} = \left[\dot{\mathbf{x}}^T, \boldsymbol{\omega}^T\right]^T \in \mathbb{R}^{3n-3}$ which appends $\dot{\mathbf{x}}$, the velocity of P in $\mathcal{F}$, with $\boldsymbol{\omega}$, the body's angular rate(s) in $\mathcal{F}$.

We now define the control problem considered in this work.

Problem 1. Assume each agent can measure the body's pose $\mathbf{q}$ and rates $\dot{\mathbf{q}}$, but has no knowledge of its mass properties (m, $\mathbf{I}_{cm}$) or geometry ($\mathbf{r}_p$, $\mathbf{r}_i$), and cannot explicitly communicate with other agents. Design a control law for each agent which allows the body to track a desired trajectory $\mathbf{q}_d(t) \in SE(n)$ asymptotically.

It is apparent that solving Problem 1 achieves collaborative manipulation under few assumptions on the information available to each agent, and their ability to communicate. However, the assumption of a common measurement $(\mathbf{q}, \dot{\mathbf{q}})$ warrants additional motivation.

At a high level, we aim to collaboratively control both the linear and rotational dynamics of the object; but in defining a reference trajectory $\mathbf{q}_d(t) \in SE(n)$, we must additionally define which point on the body we wish to track this trajectory. Since we assume the body's geometry is unknown, even if the agents can measure their own states, they have no way of computing or estimating the state of the measurement point; thus the control problem is ill-posed.

In this work, since we aim to track general trajectories on $SE(n)$, we allow the agents to simply measure the pose of the reference point; this can be achieved in practice by placing a sensor anywhere on the body, or by having a single agent broadcast its measurements to the other agents. We note this broadcast communication architecture is much simpler than a point-to-point or "mesh" network.

However, different assumptions can be made if we still wish to solve Problem 1, but without the use of a common measurement. One option is to restrict the class of desired trajectories $\mathbf{q}_d$; if the body's angular velocity $\boldsymbol{\omega} = \mathbf{0}$, then the velocity is uniform across all points on the body, and robots can use their own velocity measurements $\dot{\mathbf{x}}_i$ in order to track a desired linear velocity $\dot{\mathbf{x}}_d$. Similarly, if we only seek to control the body's orientation $\mathbf{R}$, if the agents share a reference frame, then their pose measurements are identical, making the problem again well-posed.

Finally, we can eliminate the need for a common measurement if we assume the agents know their position $\mathbf{r}_i$ with respect to the measurement point, as in [27], [28]. Using this information, the agents can easily compute the position and velocity of the reference point from their own measurements. However, we argue such an assumption is quite restrictive, since each agent must now have some way of localizing itself on the body before starting a manipulation task. We further note that such a method is not robust to errors in measuring $\mathbf{r}_i$, which could introduce unmodeled sources of uncertainty.

To solve Problem 1, we propose a modified form of the Slotine-Li adaptive manipulator controller [31], which is generalized to the multi-agent case, and can

adapt to the unknown mass and geometric parameters. We achieve tracking of a desired pose trajectory $\mathbf{q}_d(t) \in SE(3)$ by introducing a sliding variable on $SE(3)$, and show that the proposed controller indeed results in asymptotic tracking, while also ensuring all signals in the system remain bounded.

IV. Preliminaries

A. $SO(3)$ and Other Mathematical Preliminaries

In this work, we are interested in controlling both the position and orientation of a rigid body, most generally in 3D space. While, for planar tasks, we represent the body's orientation using a single angle $\theta \in [-\pi, \pi]$, orientation representation in three dimensions is more challenging.

While there exist numerous ways to parameterize the orientation of a rigid body (including Euler angles, Rodrigues parameters, and quaternions), it is well-known that three-parameter representations such as Euler angles are singular, meaning they experience configurations in which the orientation and the body's rotational dynamics are not uniquely defined. In this work, we aim to use an orientation representation which is non-singular; thus we represent orientation using a full rotation matrix $\mathbf{R} \in SO(3)$, which is both unique and non-singular.

Since $SO(3)$ is a Lie group, it admits a Lie algebra which captures the tangent space of the group at its identity element; this Lie algebra is in fact the set of skew-symmetric matrices $\mathfrak{so}(3)$, where

$$\mathfrak{so}(3) = \left\{ \mathbf{A} \in \mathbb{R}^{3\times 3} \mid \mathbf{A} = -\mathbf{A}^T \right\}.$$

We further note that $\mathfrak{so}(3)$ forms a three-dimensional vector space; thus any matrix in $\mathfrak{so}(3)$ can be mapped to a vector in $\mathbb{R}^3$ and vice versa. Let us denote by $(\cdot)^\times$ the map from $\mathbb{R}^3$ to $\mathfrak{so}(3)$, i.e. for $\mathbf{v} = [v_1, v_2, v_3]^T \in \mathbb{R}^3$,

$$\mathbf{v}^\times = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}.$$

We further denote the inverse map as $(\cdot)^\vee$.

Let us define the operator $\mathbb{P}_a : \mathbb{R}^3 \mapsto \mathbb{R}^3$, where, for $\mathbf{A} \in \mathbb{R}^{3\times 3}$,

$$\mathbb{P}_a(\mathbf{A}) = \tfrac{1}{2}\left(\mathbf{A} - \mathbf{A}^T\right),$$

which we call the "skew symmetric" part of $\mathbf{A}$. Further, let us define $\mathbb{P}_s : \mathbb{R}^3 \mapsto \mathbb{R}^3$, where

$$\mathbb{P}_s(\mathbf{A}) = \tfrac{1}{2}\left(\mathbf{A} + \mathbf{A}^T\right),$$

which we term the "symmetric" part. It is easy to verify that $\mathbb{P}_a(\mathbf{A}) \in \mathfrak{so}(3)$, and $\mathbf{A} = \mathbb{P}_a(\mathbf{A}) + \mathbb{P}_s(\mathbf{A})$.

Finally, let us define an inner product on $\mathbb{R}^{3\times 3}$,

$$\langle \mathbf{A}, \mathbf{B} \rangle = \mathrm{tr}(\mathbf{A}^T\mathbf{B}).$$

Beyond the usual properties of an inner product, it is easy to verify that $\langle \mathbb{P}_a(\mathbf{A}), \mathbb{P}_s(\mathbf{A}) \rangle = 0$, and $\langle \mathbb{P}_a(\mathbf{A}), \mathbb{P}_a(\mathbf{A}) \rangle = \frac{1}{2}||\mathbb{P}_a(\mathbf{A})^\vee||^2$.

B. Asymptotic Convergence of Nonlinear Systems

Since the topology of $SO(3)$ precludes the existence of a globally asymptotically stable equilibrium under rigid body dynamics [35], we turn to relaxed notions of stability and convergence for systems with rotational dynamics. Here we state a slightly relaxed notion of convergence, *almost global asymptotic convergence*.

Definition 1 (Almost Global Asymptotic Convergence)**.** Let $\mathbf{x}_e$ be an equilibrium of a dynamical system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$, where $\mathbf{x} \in \mathcal{X}$. We say trajectories of the system almost globally converge to x_e if, for some open, dense subset of the state space $\mathcal{A} \subset \mathcal{X}$, with closure $\bar{\mathcal{A}} = \mathcal{X}$, all trajectories with $x(0) \in \mathcal{A}$ tend towards x_e asymptotically.

V. Collaborative Manipulation as a Hamiltonian System

In this section, we will formulate the dynamics of collaborative manipulation tasks, both in the plane and in three dimensions, and show they can be represented using the Lagrangian equations for robot motion [36], commonly termed "the manipulator equations."

A. Rigid Body Dynamics in Three Dimensions

Let us consider the dynamics of a rigid body being manipulated as shown in Figure 1.

The body's linear motion is given by Newton's law,

$$\mathbf{F} = m\mathbf{a}_{cm}, \tag{1}$$

where $\mathbf{F}$ is the total force applied to the body, and $\mathbf{a}_{cm}$ is the acceleration of the body's center of mass in $\mathcal{F}$. We seek to relate $\mathbf{a}_{cm}$ to $\mathbf{a}_p$, the acceleration of P in $\mathcal{F}$. We first relate the points' velocities,

$$\mathbf{v}_p = \mathbf{v}_{cm} + \boldsymbol{\omega} \times \mathbf{R}\mathbf{r}_p,$$

where $\boldsymbol{\omega}$ is the angular velocity of $\mathcal{B}$ in $\mathcal{F}$, and $\mathbf{R}$ is the rotation matrix from $\mathcal{B}$ to $\mathcal{F}$. This can be differentiated to yield

$$\mathbf{a}_p = \mathbf{a}_{cm} + \boldsymbol{\alpha} \times \mathbf{R}\mathbf{r}_p + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{R}\mathbf{r}_p), \tag{2}$$

where $\boldsymbol{\alpha}$ is $\mathcal{B}$'s angular acceleration in $\mathcal{F}$. We can substitute (2) into (1) to yield

$$\mathbf{F} = m\mathbf{a}_p - m(\boldsymbol{\alpha} \times \mathbf{R}\mathbf{r}_p) - m\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{R}\mathbf{r}_p). \tag{3}$$

The body's rotational dynamics are given by Euler's rigid body equation,

$$\mathbf{M}_p = \mathbf{R}\mathbf{J}_p\mathbf{R}^T\boldsymbol{\alpha} + \boldsymbol{\omega} \times (\mathbf{R}\mathbf{J}_p\mathbf{R}^T\boldsymbol{\omega}) - m\mathbf{R}\mathbf{r}_p \times \mathbf{a}_p, \tag{4}$$

where $\mathbf{J}_p$ is the body's inertia matrix about $\mathbf{b}_p$, which is given by

$$\mathbf{J}_p = \mathbf{J}_{cm} + m\left((\mathbf{r}_p^T\mathbf{r}_p)\mathbf{I} - \mathbf{r}_p\mathbf{r}_p^T\right),$$

where $\mathbf{I}$ denotes the identity matrix.

We again define the configuration variable $\mathbf{q} = [\mathbf{x}, \mathbf{R}] \in \mathbb{R}^3 \times SO(3)$, where $\mathbf{x}$ is the position of the measurement point in $\mathcal{F}$. Abusing notation, we can let $\dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathbb{R}^6$, where $\dot{\mathbf{q}} = \left[\dot{\mathbf{x}}^T, \boldsymbol{\omega}^T\right]^T$ represents the body's linear and angular rates, and $\ddot{\mathbf{q}} = \left[\ddot{\mathbf{x}}^T, \boldsymbol{\alpha}^T\right]^T$ represents its linear and angular accelerations.

Using these configuration variables, we can now write (2) and (4) more compactly as

$$\boldsymbol{\tau} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}, \tag{5}$$

where $\boldsymbol{\tau}$ is the total wrench applied to the body about the measurement point, where

$$\mathbf{H}(\mathbf{q}) = \begin{bmatrix} m\mathbf{I} & m(\mathbf{R}\mathbf{r}_p)^\times \\ -m(\mathbf{R}\mathbf{r}_p)^\times & \mathbf{R}\mathbf{J}_p\mathbf{R}^T \end{bmatrix}$$

is the system's inertia matrix (which is symmetric, positive definite), and

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} \mathbf{0}_3 & m\boldsymbol{\omega}^\times(\mathbf{R}\mathbf{r}_p)^\times \\ -m\boldsymbol{\omega}^\times(\mathbf{R}\mathbf{r}_p)^\times & \boldsymbol{\omega}^\times\mathbf{R}\mathbf{I}_p\mathbf{R}^T - m((\mathbf{R}\mathbf{r}_p)^\times\dot{\mathbf{x}})^\times \end{bmatrix}$$

is a matrix which contains centrifugal and Coriolis terms, with the 3×3 zero matrix denoted by $\mathbf{0}_3$.

Further, we can show that these matrices have a number of interesting properties. Specifically, $\mathbf{H}(\mathbf{q})$ is positive definite for all $\mathbf{q}$, and $\dot{\mathbf{H}} - 2\mathbf{C}$ is skew-symmetric. These properties can be viewed as matrix versions of common properties of Hamiltonian systems (e.g. conservation of energy). Proof of these properties is included in Appendix A.

B. Grasp Matrix and its Properties

While the dynamics in (5) describe the body's motion under the total wrench $\boldsymbol{\tau}$ about $\mathbf{b}_p$, we must further express $\boldsymbol{\tau}$ as a function of each robot's applied wrench $\boldsymbol{\tau}_i$. We can write $\boldsymbol{\tau}$ as

$$\boldsymbol{\tau} = \sum_{i=1}^{N} \mathbf{M}(\mathbf{q}, \mathbf{r}_i)\boldsymbol{\tau}_i,$$

where $\mathbf{M}$ denotes the grasp matrix, which is given by

$$\mathbf{M}(\mathbf{q}, \mathbf{r}_i) = \begin{bmatrix} \mathbf{I} & \mathbf{0}_3 \\ (\mathbf{R}\mathbf{r}_i)^\times & \mathbf{I} \end{bmatrix},$$

where agent i grasps the object at some point $\mathbf{r}_i$ from the measurement point. We denote the set of such grasp matrices as $\mathcal{M}$.

The grasp matrix has some unique properties, specifically,

$$\mathbf{M}(\mathbf{r}_i)^{-1} = \begin{bmatrix} \mathbf{I} & \mathbf{0}_3 \\ -(\mathbf{R}\mathbf{r}_i)^\times & \mathbf{I} \end{bmatrix} = \mathbf{M}(-\mathbf{r}_i),$$

and

$$\begin{aligned} \mathbf{M}(\mathbf{r}_i)\mathbf{M}(\mathbf{r}_j) &= \begin{bmatrix} \mathbf{I} & \mathbf{0}_3 \\ (\mathbf{R}(\mathbf{r}_i + \mathbf{r}_j))^\times & \mathbf{I} \end{bmatrix} \\ &= \mathbf{M}(\mathbf{r}_i) + \mathbf{M}(\mathbf{r}_j) - \mathbf{I}. \end{aligned} \tag{6}$$

Thus, interestingly, for two matrices in $\mathcal{M}$, their product is simply equal to their sum, minus the identity.

C. Rigid Body Dynamics for Planar Tasks

We now turn to the case of $SE(2)$, when a group of agents seeks to manipulate a rigid body in the plane. Thus, we constrain the translational dynamics of the body in the z-dimension, and its rotational dynamics in the x- and y-dimensions to be fixed.

Accordingly, the dynamics of the planar task may also be described by (5), but with $\mathbf{q} \in \mathbb{R}^2 \times SO(2)$, $\dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathbb{R}^3$, due to the reduced degrees of freedom. Thus, enforcing the planar constraints yields

$$\mathbf{H}(\mathbf{q}) = \begin{bmatrix} m & 0 & mr_y^N \\ 0 & m & -mr_x^N \\ mr_y^N & -mr_x^N & J_p \end{bmatrix},$$

and

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} 0 & 0 & mr_x^N\omega \\ 0 & 0 & mr_y^N\omega \\ 0 & 0 & 0 \end{bmatrix},$$

where $\mathbf{r}_p^N = \left[r_x^N, r_y^N, 0\right]^T = \mathbf{R}(\theta)\mathbf{r}_p^B$, is the location of the measurement point in the global frame $\mathcal{F}$, θ is an angle which expresses the body's orientation in the global frame, ω is the body's angular velocity, and $\mathbf{R}(\theta) \in SO(2)$ is the body-to-world rotation matrix, with

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Further, when computing the total wrench $\boldsymbol{\tau}$ applied to the body, we include a frictional force $\mathbf{F}_{f,i}$ which is applied at the contact point of each robot. Thus, we can express the total wrench as

$$\boldsymbol{\tau} = \sum_{i=1}^{N} \mathbf{M}(\mathbf{q}, \mathbf{r}_i)\left(\boldsymbol{\tau}_i - \mathbf{F}_{f,i}\right),$$

where $\boldsymbol{\tau}_i$ is the wrench applied by agent i, and

$$\mathbf{M}(\mathbf{q}, \mathbf{r}_i) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -r_{i,y}^N & r_{i,x}^N & 1 \end{bmatrix}$$

is the grasp matrix for agent i, where $\mathbf{r}_i^N = \left[r_{i,x}^N, r_{i,y}^N 0\right]^T = \mathbf{R}(\theta)\mathbf{r}_i^B$ is the agent's attachment point expressed in the global frame. Thus, the full planar dynamics are given by

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \sum_{i=1}^{N} \mathbf{M}_i(\mathbf{q})\left(\boldsymbol{\tau}_i - \mathbf{F}_{f,i}\right),$$

where we omit the dependence of $\mathbf{M}_i$ on $\mathbf{r}_i$ for clarity. It is clear by inspection that these dynamics are of the same class as those in three dimensions (5).

Importantly, although the dynamics evolve in a lower-dimensional space, all properties from the full three-dimensional case hold, including the positive definiteness of $\mathbf{H}$, skew-symmetry of $\dot{\mathbf{H}} - 2\mathbf{C}$, and grasp matrix properties of $\mathbf{M}_i$.

VI. Decentralized Adaptive Trajectory Tracking

We will now propose a decentralized adaptive controller suitable for accomplishing the collaborative manipulation tasks outlined previously, and provide proofs of boundedness of signals, and almost global asymptotic tracking.

A. Proposed Sliding Variable on $SE(3) \times \mathbb{R}^6$

We begin by proposing a sliding surface for 3D poses and their derivatives, which lie in $SE(3) \times \mathbb{R}^6$. Adaptive controllers for Hamiltonian manipulators in literature (cf. [37, Chapter 9]) typically utilize sliding surfaces which correspond to a linear subspace, and thus have linear error dynamics on the surface. Such a surface is appropriate for configuration variables (such as positions and small joint angles) which lie in Cartesian space.

However, since we aim to track both a desired position $\mathbf{x}_d$ and orientation $\mathbf{R}_d$, the system's dynamics do not evolve naturally in Cartesian space. We thus propose a sliding variable $\mathbf{s}$ which is non-singular, inherits the topology of $SE(3)$, and exhibits almost global convergence when on the sliding surface $\mathcal{D} = \left\{(\mathbf{q}_e, \dot{\mathbf{q}}_e) \in SE(3) \times \mathbb{R}^6 \mid \mathbf{s}(\mathbf{q}_e, \dot{\mathbf{q}}_e) = 0\right\}$. To achieve this, we borrow the sliding variable on $SO(3)$ from [38], and combine it with a typical linear sliding variable on $\mathbb{R}^3$ to define a sliding surface on $SE(3)$. We then use a Lyapunov-like proof to show almost global convergence to tracking of the desired orientation trajectory when on the surface.

We aim to control the body to track a desired position $\mathbf{x}_d(t)$, and a desired orientation, $\mathbf{R}_d$, which satsifies

$$\dot{\mathbf{R}}_d = \boldsymbol{\omega}_d^{\times} \mathbf{R}_d,$$

for some desired angular velocity $\boldsymbol{\omega}_d$. Let us define the rotational and angular velocity errors as $\mathbf{R}_e = \mathbf{R}_d^T \mathbf{R}$, and $\boldsymbol{\omega}_e = \boldsymbol{\omega} - \boldsymbol{\omega}_d$, noting

$$(\mathbf{R}_e, \boldsymbol{\omega}_e) = (\mathbf{I}, \mathbf{0}) \iff (\mathbf{R}, \boldsymbol{\omega}) = (\mathbf{R}_d, \boldsymbol{\omega}_d).$$

We can further obtain the rotation error dynamics by taking a time derivative of $\mathbf{R}_e$, yielding

$$\begin{aligned} \dot{\mathbf{R}}_e &= \dot{\mathbf{R}}_d^T \mathbf{R} + \mathbf{R}_d^T \dot{\mathbf{R}}, \\ &= \mathbf{R}_d^T (\boldsymbol{\omega} - \boldsymbol{\omega}_d)^{\times} \mathbf{R}, \\ &= \left(\mathbf{R}_d^T (\boldsymbol{\omega} - \boldsymbol{\omega}_d)\right)^{\times} \mathbf{R}_d^T \mathbf{R}, \\ &= \left(\mathbf{R}_d^T \boldsymbol{\omega}_e\right)^{\times} \mathbf{R}_e. \end{aligned} \tag{7}$$

Finally, we propose a sliding variable $\mathbf{s} : SE(3) \times \mathbb{R}^6 \mapsto \mathbb{R}^6$, which is given by

$$\mathbf{s} = \begin{bmatrix} \mathbf{s}_\ell \\ \boldsymbol{\sigma} \end{bmatrix} = \begin{bmatrix} \dot{\tilde{\mathbf{x}}} + \lambda \tilde{\mathbf{x}} \\ \boldsymbol{\omega}_e + \lambda \mathbf{R}_d \mathbb{P}_a(\mathbf{R}_e)^{\vee} \end{bmatrix},$$

where $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_d$ is the linear tracking error, and $\mathbf{s}_\ell : \mathbb{R}^3 \times \mathbb{R}^3 \mapsto \mathbb{R}^3$ and $\boldsymbol{\sigma} : SO(3) \times \mathbb{R}^3 \mapsto \mathbb{R}^3$ are sliding variables for the object's linear and angular dynamics, respectively.

B. System Behavior on Sliding Surface

We now consider the system's behavior when it lies on the sliding surface $\mathbf{s} = 0$. The behavior of the linear dynamics is obvious; imposing the condition $\mathbf{s}_\ell = 0$ results in the linear error dynamics

$$\dot{\tilde{\mathbf{x}}} = -\lambda \tilde{\mathbf{x}},$$

which defines a stable linear system. Thus, once on the sliding surface, the position tracking error $\tilde{\mathbf{x}}$ tends exponentially to zero, with time constant $\frac{1}{\lambda}$.

Thus, we turn our attention to the rotational error dynamics which are defined by $\boldsymbol{\sigma} = 0$. Imposing this condition results in the rotational dynamics

$$\begin{aligned} \dot{\mathbf{R}}_e &= \left(-\lambda \mathbf{R}_d^T \mathbf{R}_d \mathbb{P}_a(\mathbf{R}_e)^{\vee}\right)^{\times} \mathbf{R}_e, \\ &= -\lambda \mathbb{P}_a(\mathbf{R}_e) \mathbf{R}_e, \end{aligned}$$

using the rotation error dynamics given by (7). We note that $(\mathbf{R}_e, \boldsymbol{\omega}_e) = (\mathbf{I}, \mathbf{0})$ both lies on the sliding surface, and is an equilibrium of the reduced-order rotational dynamics.

Theorem 1. The rotation error $(\mathbf{R}_e, \boldsymbol{\omega}_e)$ almost globally converges to the equilibrium $(\mathbf{I}, \mathbf{0})$ for the reduced-order system given by $\boldsymbol{\sigma} = 0$.

Proof. Consider the Lyapunov function candidate

$$V_R(t) = \operatorname{tr}(\mathbf{I} - \mathbf{R}_e) > 0,$$

which is positive definite since $-1 \leq \operatorname{tr}(\mathbf{R}_e) \leq 3$ for all $\mathbf{R}_e \in SO(3)$, with $\operatorname{tr}(\mathbf{R}_e) = 3$ iff $\mathbf{R}_e = \mathbf{I}$. The time derivative of V_R is given by

$$\begin{aligned} \dot{V}_R &= -\operatorname{tr}(\dot{\mathbf{R}}_e), \\ &= \lambda \operatorname{tr}(\mathbb{P}_a(\mathbf{R}_e)\mathbf{R}_e), \\ &= \lambda \left\langle -\mathbb{P}_a(\mathbf{R}_e), \mathbf{R}_e \right\rangle, \\ &= -\lambda \left\langle \mathbb{P}_a(\mathbf{R}_e), \mathbb{P}_a(\mathbf{R}_e) + \mathbb{P}_s(\mathbf{R}_e) \right\rangle, \\ &= -\tfrac{\lambda}{2} ||\mathbb{P}_a(\mathbf{R}_e)^{\vee}||^2 \leq 0, \end{aligned}$$

using the matrix inner product $\langle \mathbf{A}, \mathbf{B} \rangle = \mathrm{tr}(\mathbf{A}^T \mathbf{B})$, and the fact that $\langle \mathbb{P}_a(\mathbf{A}), \mathbb{P}_s(\mathbf{A}) \rangle = 0$, for $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{3\times 3}$.

Further, invoking Barbalat's lemma, [37, Chapter 4.5], we can conclude that if V_R has a finite limit, and if $\dot{V}_R$ is uniformly continuous, the rotational error will converge to the set

$$\mathcal{E} = \left\{ \mathbf{R}_e \mid \dot{V}_R = 0 \right\}.$$

Since V_R is lower bounded, and $\dot{V}_R$ is negative semi-definite, it must have a finite limit. Thus, we show $\dot{V}_R$ is uniformly continuous.

To show $\dot{V}_R$ is uniformly continuous, it is sufficient to show $\ddot{V}_R$ is bounded. To this end, we can differentiate $\dot{V}_R$ to yield

$$\begin{aligned} \ddot{V}_R &= -\lambda \left\langle \mathbb{P}_a(\mathbf{R}_e), \mathbb{P}_a(\dot{\mathbf{R}}_e) \right\rangle \\ &= \frac{\lambda^2}{2} \left\langle \mathbb{P}_a(\mathbf{R}_e), \mathbb{P}_a(\mathbf{R}_e \mathbf{R}_e) \right\rangle, \\ &= \frac{\lambda^2}{2} \mathrm{tr}\left(\mathbf{R}_e - \mathbf{R}_e^3 + \mathbf{R}_e^T - (\mathbf{R}_e^T)^3 \right), \end{aligned}$$

which must be bounded, since all matrices inside the trace lie in $SO(3)$, and must have a trace in $[-1, 3]$. Thus, we must have $\dot{V}_R$ uniformly continuous, which implies $\mathbf{R}_e \to \mathcal{E}$.

Again using the results from [38], we find that

$$\mathcal{E} = \{\mathbf{I}\} \cup \left\{ \mathbf{I} + 2\boldsymbol{\eta}^\times \boldsymbol{\eta}^\times \mid ||\boldsymbol{\eta}|| = 1 \right\}.$$

Further, since the spurious points in $\mathcal{E}$ correspond to maxima of V_R, all points in $\mathcal{E}$ are unstable except for $\mathbf{R}_e = \mathbf{I}$. Thus, since the spurious equilibria are unstable, and lie in a nowhere dense set, we conclude $(\mathbf{R}_e, \boldsymbol{\omega}_e)$ almost globally converges to $(\mathbf{I}, \mathbf{0})$. □

Thus, we can conclude that when the system lies on the sliding surface $\mathbf{s} = 0$, it will almost globally converge to tracking the desired position $\mathbf{x}_d$ and orientation $\mathbf{R}_d$.

We further note that $\mathbf{s}$ can be interpreted as a "velocity error," since we can write

$$\mathbf{s} = \dot{\mathbf{q}} - \dot{\mathbf{q}}_r,$$

where $\dot{\mathbf{q}}_r$ is a "reference velocity" which augments the desired velocity $\dot{\mathbf{q}}_d$ with an additional term based on the pose error,

$$\dot{\mathbf{q}}_r = \begin{bmatrix} \dot{\mathbf{x}}_d - \lambda \widetilde{\mathbf{x}} \\ \boldsymbol{\omega}_d - \lambda \mathbf{R}_d \mathbb{P}_a(\mathbf{R}_e)^\vee \end{bmatrix}.$$

C. Decentralized Adaptive Trajectory Control

We now consider the problem of decentralized adaptive control of a common payload. We consider Lagrangian dynamics of the form

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g} = \sum_{i=1}^{N} \mathbf{M}_i(\mathbf{q}) \boldsymbol{\tau}_i, \tag{8}$$

which, as shown previously, hold for collaborative manipulation tasks on both $SE(2)$ and $SE(3)$. We note that while neither of the studied systems includes the term $\mathbf{g}$, which models conservative forces such as gravity, we include it here for completeness, and to conform to the traditional "manipulator equation."

We now consider a Lyapunov function candidate of the form

$$V(t) = \tfrac{1}{2} \left[\mathbf{s}^T \mathbf{H} \mathbf{s} + \textstyle\sum_{i=1}^{N} \widetilde{\mathbf{o}}_i^T \boldsymbol{\Gamma}_o^{-1} \widetilde{\mathbf{o}}_i + \widetilde{\mathbf{r}}_i^T \boldsymbol{\Gamma}_r^{-1} \widetilde{\mathbf{r}}_i \right], \tag{9}$$

where $\widetilde{\mathbf{o}}_i = \hat{\mathbf{o}}_i - \mathbf{o}_i$ is a parameter estimation error for agent i, with $\mathbf{o}_i$ being a vector of physical constants that define the payload's dynamics, and $\hat{\mathbf{o}}_i$ being agent i's estimate of these parameters. Similarly, we let $\widetilde{\mathbf{r}}_i$ be the difference between the vector $\mathbf{r}_i$ from the measurement point to agent i, and the agent's estimate $\hat{\mathbf{r}}_i$. We further let $\boldsymbol{\Gamma}_o$ and $\boldsymbol{\Gamma}_r$ be positive definite matrices which correspond to adaptation gains.

Differentiating V yields

$$\dot{V}(t) = \mathbf{s}^T \mathbf{H} \dot{\mathbf{s}} + \tfrac{1}{2} \mathbf{s}^T \dot{\mathbf{H}} \mathbf{s} + \textstyle\sum_{i=1}^{N} \widetilde{\mathbf{o}}_i^T \boldsymbol{\Gamma}_o^{-1} \dot{\hat{\mathbf{o}}} + \widetilde{\mathbf{r}}_i^T \boldsymbol{\Gamma}_r^{-1} \dot{\hat{\mathbf{r}}}_i,$$

using the fact that the parameter error derivatives $\dot{\widetilde{\mathbf{o}}}_i, \dot{\widetilde{\mathbf{r}}}_i$ are equal to the parameter estimate derivatives $\dot{\hat{\mathbf{o}}}_i, \dot{\hat{\mathbf{r}}}_i$, since the true parameter values are constant.

Further, since we can write $\dot{\mathbf{s}} = \ddot{\mathbf{q}} - \ddot{\mathbf{q}}_r$, we can expand the term

$$\begin{aligned} \mathbf{s}^T \mathbf{H} \dot{\mathbf{s}} &= \mathbf{s}^T \mathbf{H} (\ddot{\mathbf{q}} - \ddot{\mathbf{q}}_r), \\ &= \mathbf{s}^T \left(\textstyle\sum_{i=1}^{N} \mathbf{M}_i \boldsymbol{\tau}_i - \mathbf{C}\dot{\mathbf{q}} - \mathbf{g} - \mathbf{H}\ddot{\mathbf{q}}_r \right), \\ &= \mathbf{s}^T \left(\textstyle\sum_{i=1}^{N} \mathbf{M}_i \boldsymbol{\tau}_i - \mathbf{H}\ddot{\mathbf{q}}_r - \mathbf{C}\dot{\mathbf{q}}_r - \mathbf{g} - \mathbf{C}\mathbf{s} \right), \end{aligned}$$

using the system dynamics (8) to substitute for $\mathbf{H}\ddot{\mathbf{q}}$, and the fact that $\dot{\mathbf{q}} = \mathbf{s} + \dot{\mathbf{q}}_r$.

Further, since the system is over-actuated, meaning it has many more control inputs than degrees of freedom, there exist many sets of control inputs which produce the same object dynamics. Thus, it is sufficient for each agent to take some portion of the control effort, which, combined, enables tracking of the desired trajectory. To this end, let us consider a set of N positive constants α_i, with $\sum_{i=1}^{N} \alpha_i = 1$. If we again consider the analogy between our proposed controller and quorum sensing [33], we can interpret these constants as akin to coupling weights between the agents.

Using these constants, we can allocate the total control effort needed for feedback linearization among the agents, writing

$$\mathbf{M}_i \boldsymbol{\tau}_i = \alpha_i \left(\mathbf{H}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{g} \right)$$

for each i. It is important to note that these constants, in effect, only scale the physical parameters of the system, meaning they can be adapted to and do not need to be known *a priori*.

We can also exploit an important property of this system, *linear parameterizability*. Importantly, for the control tasks considered in this work, the physical parameters to be estimated appear only linearly in the dynamics. Thus, we can define a known regressor matrix $\mathbf{Y}_o(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)$, such that

$$\alpha_i \left(\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}}_r + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_r + \mathbf{G}\right) = \mathbf{Y}_o(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)\mathbf{o}_i.$$

We further note that

$$\begin{aligned}\textstyle\sum_{i=1}^N \mathbf{Y}_o\mathbf{o}_i &= \textstyle\sum_{i=1}^N \alpha_i \left(\mathbf{H}\ddot{\mathbf{q}}_r + \mathbf{C}\dot{\mathbf{q}}_r + \mathbf{g}\right)\\ &= \mathbf{H}\ddot{\mathbf{q}}_r + \mathbf{C}\dot{\mathbf{q}}_r + \mathbf{g}.\end{aligned}$$

Using this regressor, we can write

$$\begin{aligned}\dot{V}(t) = &\textstyle\sum_{i=1}^N \mathbf{s}^T \left(\mathbf{M}_i\boldsymbol{\tau}_i - \mathbf{Y}_o\mathbf{o}_i\right) + \tfrac{1}{2}\mathbf{s}^T\left(\dot{\mathbf{H}} - 2\mathbf{C}\right)\mathbf{s}\\ &+ \widetilde{\mathbf{o}}_i^T\boldsymbol{\Gamma}_o^{-1}\dot{\hat{\mathbf{o}}} + \widetilde{\mathbf{r}}_i^T\boldsymbol{\Gamma}_r^{-1}\dot{\hat{\mathbf{r}}}_i. \end{aligned} \tag{10}$$

Finally, since as shown previously, $\dot{\mathbf{H}} - 2\mathbf{C}$ is skew-symmetric, its quadratic form is equal to zero for any vector $\mathbf{s}$.

We will now outine the proposed control and adaptation laws. Let

$$\boldsymbol{\tau}_i = \widehat{\mathbf{M}}_i^{-1}\mathbf{F}_i, \tag{11}$$

where $\mathbf{M}_i(\hat{\mathbf{r}}_i)^{-1}$ is the inverse of a grasp matrix in $\mathcal{M}$ with a moment arm of $\hat{\mathbf{r}}_i$, and

$$\mathbf{F}_i = \mathbf{Y}_o\hat{\mathbf{o}}_i - \mathbf{K}_D\mathbf{s}, \tag{12}$$

where $\mathcal{F}_D$ is some positive definite matrix. Intuitively, the term $\mathbf{F}_i$ combines a feedforward term using the estimated object dynamics with a simple PD term, $-\mathbf{K}_D\mathbf{s}$. This control law alone would lead to successful tracking if all agents were attached at the measurement point. Each agent then multiplies $\mathbf{F}_i$ by $\widehat{\mathbf{M}}_i^{-1}$ in an effort to compensate for the torque its applied force generates about the measurement point.

We further note, using (6), that the product

$$\mathbf{M}_i\widehat{\mathbf{M}}_i^{-1} = \left(\mathbf{I} - \widetilde{\mathbf{M}}_i\right),$$

where $\widetilde{\mathbf{M}}_i = \widehat{\mathbf{M}}_i - \mathbf{M}_i$. We can further observe that the product $\widetilde{\mathbf{M}}_i\mathbf{F}_i$ is linearly parameterizable, allowing us to write

$$-\widetilde{\mathbf{M}}_i\mathbf{F}_i = \mathbf{Y}_g(\mathbf{F}_i, \mathbf{q})\widetilde{\mathbf{r}}_i.$$

Using this expression, we now consider the adaptation laws

$$\dot{\hat{\mathbf{o}}}_i = -\Gamma_o\mathbf{Y}_o(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)^T\mathbf{s}, \tag{13}$$

$$\dot{\hat{\mathbf{r}}}_i = -\Gamma_r\mathbf{Y}_g(\mathbf{F}_i, \mathbf{q})^T\mathbf{s}. \tag{14}$$

Note that all terms in the proposed control and adaptation laws can be computed using only local information (using the desired trajectory, shared measurement, and local parameter estimates). We now reach the main theoretical result of this paper, namely the boundedness of all closed-loop signals, and asymptotic convergence of the system trajectory to the desired trajectory.

Theorem 2. Consider the control laws (11)-(12), and the adaptation laws (13)-(14). Under the proposed adaptive controller, all closed-loop signals remain bounded, and the system converges to the desired trajectory from almost all initial conditions.

Proof. Substituting these control and adaptation laws into the previous expression (10) of $\dot{V}(t)$ yields

$$\begin{aligned}\dot{V}(t) = &\textstyle\sum_{i=1}^N \mathbf{s}^T\left(\mathbf{F}_i + \mathbf{Y}_g\widetilde{\mathbf{r}}_i - \mathbf{Y}_o\mathbf{o}_i\right)\\ &+ \widetilde{\mathbf{o}}_i^T\boldsymbol{\Gamma}_o^{-1}\dot{\hat{\mathbf{o}}} + \widetilde{\mathbf{r}}_i^T\boldsymbol{\Gamma}_r^{-1}\dot{\hat{\mathbf{r}}}_i,\\ = &\textstyle\sum_{i=1}^N -\mathbf{s}^T\mathbf{K}_D\mathbf{s} + \widetilde{\mathbf{o}}_i^T\left(\mathbf{Y}_o\widetilde{\mathbf{o}}_i + \boldsymbol{\Gamma}_o^{-1}\dot{\hat{\mathbf{o}}}_i\right)\\ &+ \widetilde{\mathbf{r}}^T\left(\mathbf{Y}_g\widetilde{\mathbf{r}}_i + \boldsymbol{\Gamma}_r^{-1}\dot{\hat{\mathbf{r}}}_i\right),\end{aligned}$$

which, upon substituting the adaptation laws (13)-(14), yields

$$\dot{V}(t) = N\left(-\mathbf{s}^T\mathbf{K}_D\mathbf{s}\right) \leq 0. \tag{15}$$

From the previous results in Section VI-B, we know that $\mathbf{s} \to \mathbf{0}$ is sufficient to show that the body tracks the desired trajectory asymptotally. Thus, we now show that $\dot{V}(t) \to 0$ as $t \to \infty$, since from (15) it is apparent that this is sufficient for $\mathbf{s} \to \mathbf{0}$.

To do this, we again invoke Barbalat's Lemma [37, Chapter 4.5], which states that $\dot{V} \to 0$ if V has a finite limit, and $\dot{V}$ is uniformly continuous. It is apparent again from (15) that V has a finite limit; thus, we show $\ddot{V}$ is bounded.

From the expression of $\ddot{V} = -2N\left(\mathbf{s}^T\mathbf{K}_D\dot{\mathbf{s}}\right)$, we have $\ddot{V}$ is bounded if $\mathbf{s}, \dot{\mathbf{s}}$ are bounded. Since $\dot{V} \leq 0$, and V is lower-bounded, we have that V is bounded, which implies $\mathbf{s}, \widetilde{\mathbf{o}}_i, \widetilde{\mathbf{r}}_i$ are bounded for all i.

Further, we can write

$$\begin{aligned}\mathbf{H}\dot{\mathbf{s}} &= \mathbf{H}(\ddot{\mathbf{q}} - \ddot{\mathbf{q}}_r),\\ &= \textstyle\sum_{i=1}^N \left(\mathbf{F}_i + \mathbf{Y}_g\widetilde{\mathbf{r}}_i\right) - \mathbf{Y}_o\mathbf{o}_i - \mathbf{C}\mathbf{s},\\ &= \textstyle\sum_{i=1}^N \left(\mathbf{Y}_o\widetilde{\mathbf{o}}_i + \mathbf{Y}_g\widetilde{\mathbf{r}}_i\right) - \left(N\mathbf{K}_D + \mathbf{C}\right)\mathbf{s},\end{aligned}$$

noting that all signals on the right-hand side are bounded (since the desired trajectory $\mathbf{q}_d$ and its derivatives are bounded by design). Finally, since $\mathbf{H}$ is positive definite, we know $\mathbf{H}^{-1}$ exists, which implies that $\dot{\mathbf{s}}$ is bounded.

Thus, we have shown all signals are bounded. Further, since $\mathbf{s} \to \mathbf{0}$ as $t \to \infty$, then we have $\dot{\widetilde{\mathbf{q}}}, \widetilde{\mathbf{x}} \to \mathbf{0}, \mathbf{R}_d^T\mathbf{R} \to \mathbf{I}$ in the limit, which completes the proof. □

D. Lack of Persistent Excitation

While we have shown that under the proposed controller, the tracking errors approach zero in the limit, we have not reasoned about the asymptotic behavior of the parameter errors $\widetilde{\mathbf{o}}_i, \widetilde{\mathbf{r}}_i$. While we know these signals

remain bounded, we must still investigate under what conditions the parameter estimates converge to their true values.

For most (single-agent) adaptive controllers, an additional condition termed "persistent excitation" ensures that the parameter errors are driven to zero asymptotically.

Definition 2 (Persistent Excitation)**.** Consider $\mathbf{W}(t) \in \mathbb{R}^{m\times n}$, a matrix-valued time-varying signal. We say $\mathbf{W}(t)$ is persistently exciting if, for all $t > 0$, there exist $\gamma, T > 0$ such that

$$\frac{1}{T}\int_t^{t+T} \mathbf{W}(\tau)^T\mathbf{W}(\tau)d\tau > \gamma\mathbf{I}.$$

Let us first consider the case of single-agent manipulation, i.e. the case where $N = 1$. Further, let us define $\mathbf{Y}_1 = [\mathbf{Y}_o, \mathbf{Y}_r]$, and $\widetilde{\mathbf{a}}_1 = \left[\widetilde{\mathbf{o}}^T, \widetilde{\mathbf{r}}^T\right]^T$, meaning we stack the parameter errors row-wise, and their respective regressors column-wise.

In this case, we can again write the closed-loop dynamics as

$$\mathbf{H}\dot{\mathbf{s}} + (\mathbf{C} + \mathbf{K}_D)\,\mathbf{s} = \mathbf{Y}_1\widetilde{\mathbf{a}}_1,$$

which, since we know $\mathbf{s}, \dot{\mathbf{s}} \to \mathbf{0}$ in the limit, implies that $\widetilde{\mathbf{a}}_1$ converges to the nullspace $\mathcal{F}(\mathbf{Y}_1)$ of $\mathbf{Y}_1$; this does not, however, imply that $\widetilde{\mathbf{a}}_1 \to \mathbf{0}$.

Instead, it is sufficient to require that the matrix $\mathbf{Y}_{1,d}(t)$ is persistently exciting [39], where $\mathbf{Y}_{1,d}(t) = \mathbf{Y}_1(\mathbf{q}_d, \dot{\mathbf{q}}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d)$. However, while there exist simple conditions on the reference signal to guarantee persistent excitation for linear, time-invariant systems, to date, such simple conditions do not exist for nonlinear systems. In practice, however, reference signals with a large amount of frequency content are typically used to achieve persistent excitation.

We now turn to the multi-agent case. Let us define $\mathbf{Y}_N = [\mathbf{Y}_o, \mathbf{Y}_r, \cdots, \mathbf{Y}_o, \mathbf{Y}_r]$ and $\widetilde{\mathbf{a}}_N = \left[\widetilde{\mathbf{o}}_1^T, \widetilde{\mathbf{r}}_1^T, \cdots, \widetilde{\mathbf{o}}_N^T, \widetilde{\mathbf{r}}_N^T\right]^T$, i.e. we stack $\mathcal{F}$ sets of the regressors $\mathbf{Y}_o, \mathbf{Y}_r$ column-wise, and append all $\mathcal{F}$ sets of parameter errors row-wise. This allows us to again write the closed-loop dynamics as

$$\mathbf{H}\dot{\mathbf{s}} + (\mathbf{C} + \mathbf{K}_D)\,\mathbf{s} = \mathbf{Y}_N\widetilde{\mathbf{a}}_N.$$

Thus, to ensure that all parameter errors $\widetilde{\mathbf{a}}_i$ converge to zero, we hope to find conditions under which $\mathbf{Y}_{N,d}(\mathbf{q}_d, \dot{\mathbf{q}}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d)$ is persistently exciting. Thus, for all $t > 0$, we require that there exist $T, \gamma > 0$ such that

$$\int_t^{t+T} \mathbf{Y}_{N,d}^T(\tau)\mathbf{Y}_{N,d}(\tau)d\tau \geq \gamma\mathbf{I}.$$

However, examining the structure of the inner product $\mathbf{Y}_{N,d}^T\mathbf{Y}_{N,d}$ reveals

$$\mathbf{Y}_{N,d}^T(t)\mathbf{Y}_{N,d}(t) = \begin{bmatrix} \mathbf{Y}_o^T\mathbf{Y}_o & \mathbf{Y}_o^T\mathbf{Y}_r & \cdots & \mathbf{Y}_o^T\mathbf{Y}_r \\ \mathbf{Y}_r^T\mathbf{Y}_o & \mathbf{Y}_r^T\mathbf{Y}_r & \cdots & \mathbf{Y}_r^T\mathbf{Y}_r \\ \vdots & & \ddots & \vdots \\ \mathbf{Y}_r^T\mathbf{Y}_o & \mathbf{Y}_r^T\mathbf{Y}_r & \cdots & \mathbf{Y}_r^T\mathbf{Y}_r \end{bmatrix},$$

which we note has the same two (block) rows repeated $\mathcal{F}$ times. Further, we note the integral

$$\int_t^{t+T} \mathbf{Y}_{N,d}^T(\tau)\mathbf{Y}_{N,d}(\tau)d\tau$$

will have the same block structure, meaning the integrated matrix must be rank deficient.

Thus, regardless of the reference signal $\mathbf{q}_d$, $\mathbf{Y}_{N,d}$ cannot be persistently exciting; therefore we can make no claim on the asymptotic behavior of the individual parameter errors $\widetilde{\mathbf{a}}_i$, except that they remain bounded, and their derivative $\dot{\widetilde{\mathbf{a}}}_i$ converges to zero.

However, we can also write the closed-loop dynamics as

$$\begin{aligned}\mathbf{H}(\mathbf{q})\dot{\mathbf{s}} + (\mathbf{C} + \mathbf{K}_D)\mathbf{s} &= \textstyle\sum_{i=1}^N \mathbf{Y}_1\widetilde{\mathbf{a}}_i, \\ &= N\mathbf{Y}_1\left(\tfrac{1}{N}\textstyle\sum_{i=1}^N \widetilde{\mathbf{a}}_i\right),\end{aligned}$$

which implies, if $\mathbf{Y}_1$ is persistently exciting, that the average parameter error $\frac{1}{N}\sum_{i=1}^N \widetilde{\mathbf{a}}_i$ converges to zero.

Thus, we can interpret the lack of a persistently exciting signal $\mathbf{q}_d$ as a reflection of the redundancy inherent in the system; since there exists a subspace of individual wrenches $\boldsymbol{\tau}_i$ which produce the same total wrench $\boldsymbol{\tau}$ on the body, even a signal which would be persistently exciting for a single agent can only guarantee that the average parameter error vanishes.

Interestingly, this is in many ways dual to the results in [40], wherein the authors consider the problem of adaptive control for cloud-based robots. In this case, the authors have a large number of decoupled, but identical, manipulators, which they aim to control adaptively using a shared parameter estimate. The authors find that parameter convergence is guaranteed if the sum of the regressor matrices is persistently exciting, a much weaker condition than any single robot executing a persistently exciting trajectory. On the other hand, in this setting, we have a single physical system which is controlled collaboratively by many robots using individual parameter estimates; we find that in this case persistent excitation requires not just a stronger condition on $\mathbf{q}_d$, but is in fact impossible to achieve.

E. Friction and Dissipative Dynamics

Finally, in some systems (such as the planar manipulation task), there are non-conservative forces which

act on the payload due to effects such as friction. The simplest model treats these forces as an additive body-frame disturbance $\mathbf{D}(\mathbf{q})\dot{\mathbf{q}}$ which is linear in the velocity of the measurement point, where $\mathbf{D}(\mathbf{q}) = \mathbf{R}\boldsymbol{\Lambda}_D\mathbf{R}^T$, for a constant positive semidefinite matrix of friction coefficients $\boldsymbol{\Lambda}_D$. It can be easily verified that such a model is appropriate for viscous sliding friction.

This friction can be compensated quite easily using an additional term in the control law, namely

$$\mathbf{F}_i = \mathbf{Y}_o\hat{\mathbf{o}}_i + \mathbf{Y}_f\hat{\mathbf{f}}_i - \mathbf{K}_D\mathbf{s},$$

where $\mathbf{Y}_f(\mathbf{q}, \dot{\mathbf{q}}_r)\mathbf{f}_i = \alpha_i\mathbf{D}(\mathbf{q})\dot{\mathbf{q}}_r$. If we additionally use the adaptation law

$$\dot{\hat{\mathbf{f}}}_i = -\boldsymbol{\Gamma}_f\mathbf{Y}_f^T(\mathbf{q}, \dot{\mathbf{q}}_r)\mathbf{s},$$

with positive definite $\boldsymbol{\Gamma}_D$, one can show this results in

$$\dot{V} = -\mathbf{s}^T\left(N\mathbf{K}_D + \mathbf{D}(\mathbf{q})\right)\mathbf{s},$$

recalling that $\mathbf{D}\mathbf{s} = \mathbf{D}\dot{\mathbf{q}} - \mathbf{D}\dot{\mathbf{q}}_r$. Thus, in effect, we have increased the PD gain with an additional positive semi-definite term $\mathbf{D}(\mathbf{q})$, yielding a higher "effective" $\mathbf{K}_D$ matrix without increasing the gains directly.

However, since such an expression is not obtainable in general, e.g. in the case of Coulomb friction, we turn instead to the case when frictional forces are modeled as being applied at the attachment point of each agent. In this case, the frictional forces are also pre-multiplied by the grasp matrices, resulting in the dynamics

$$\mathbf{H}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{g} = \textstyle\sum_{i=1}^N \mathbf{M}_i\left(\boldsymbol{\tau}_i - \mathbf{D}\mathbf{v}_i - \mathbf{D}_C\operatorname{sgn}(\mathbf{v}_i)\right),$$

where $\mathbf{v}_i = \mathbf{M}_i^T\dot{\mathbf{q}}$ is the linear and angular velocity of the robot in the global frame, $\mathbf{D}$ and $\mathbf{D}_C$ are positive semidefinite matrices of friction coefficients, and $\operatorname{sgn}(\cdot)$ is the element-wise sign function.

These terms can again be compensated, now by augmenting the "shifted" control law (11),

$$\boldsymbol{\tau}_i = \widehat{\mathbf{M}}_i^{-1}\mathbf{F}_i + \mathbf{Y}_D\hat{\mathbf{d}}_i + \mathbf{Y}_c\hat{\mathbf{c}}_i,$$

where we have

$$\mathbf{Y}_D(\mathbf{q}, \dot{\mathbf{q}}_r)\mathbf{d}_i = \mathbf{M}_i\mathbf{D}\mathbf{M}_i^T\dot{\mathbf{q}}_r,$$

and

$$\mathbf{Y}_C(\mathbf{q}, \mathbf{v}_i)\mathbf{c}_i = \mathbf{M}_i\mathbf{D}_C\operatorname{sgn}(\mathbf{v}_i),$$

which can be computed assuming each agent can measure its own linear and angular rate $\mathbf{v}_i$.

Using this new control law, along with adaptation laws

$$\dot{\hat{\mathbf{d}}}_i = -\boldsymbol{\Gamma}_D\mathbf{Y}_D^T\mathbf{s},$$
$$\dot{\hat{\mathbf{c}}}_i = -\boldsymbol{\Gamma}_C\mathbf{Y}_C^T\mathbf{s},$$

results in

$$\dot{V} = -\mathbf{s}^T\left(N\mathbf{K}_D + \mathbf{M}_i\mathbf{D}\mathbf{M}_i^T\right)\mathbf{s},$$

which again adds a positive definite term $\mathbf{M}_i\mathbf{D}\mathbf{M}_i^T$ to $\dot{V}$, using the viscous friction applied at each point. We note that the Coulomb friction must be cancelled exactly, due to the presence of unknown parameters (namely $\mathbf{r}_i$) in the $\operatorname{sgn}(\cdot)$ nonlinearity.

VII. Simulation Results

We simulated the performance of the proposed controller for a collaborative manipulation task in $SE(3)$ to verify the previous theoretical results. To this end, we constructed a simulation scenario where a group of $N = 6$ space robots are controlling a large, cylindrical rigid body (such as a rocket body) to track a desired periodic trajectory $\mathbf{q}_d(t)$.

Table I lists the parameters used in the simulation, along with their values. The payload was a cylinder with a height of 1.5 m and radius 0.5 m. The agents were placed symmetrically along the principal axes of the body, with the measurement point $\mathbf{r}_p$ collocated with the first agent, $\mathbf{r}_1$, replicating the scenario when one agent broadcasts its measurements to the group. The reference trajectory $\mathbf{q}_d$ had desired linear and angular rates $\dot{\mathbf{q}}_d$ which were sums of sinusoids, e.g.

$$v_{x,d}(t) = \sum_{i=1}^{n_f}\alpha_i\cos(\omega_i t),$$

with the number of frequencies $n_f = 5$. The desired pose $\mathbf{q}_d$ started from an initial state $\mathbf{q}_d(0)$ and forward-integrated the dynamics according to the desired velocity.

The simulations were implemented in Julia 1.1.0. All continuous dynamics were discretized and numerically integrated using Heun's method [41], a two-stage Runge-Kutta method, using a step size of $h = 1 \times 10^{-2}$ s. All forward integration steps involving rotation matrices were implemented using the so-called "exponential map"

$$\mathbf{R}_{t+h} = \exp\left(h\boldsymbol{\omega}_t^\times\right)\mathbf{R}_t,$$

which ensures that $\mathbf{R}$ remains in $SO(3)$ for all time. Since the proposed adaptation laws are designed for continuous-time systems, we modify them by adding a deadband in which no adaptation will occur. For all simulations, we stop adaptation when $||\mathbf{s}||_2 \leq 0.01$. This technique is well-studied, and is commonly used to improve the robustness of adaptive controllers to unmodeled dynamics.

The physical parameters of the rigid body are given in Table I. The gain matrix $\boldsymbol{\Gamma}_o$ was set to $0.3 * \operatorname{diag}(|\mathbf{a}_o| + 0.01)$, where, for a vector $\mathbf{v} \in \mathbb{R}^n$, $\operatorname{diag}(\mathbf{v})$ returns the matrix in $\mathbb{R}^{n\times n}$ with $\mathbf{v}$ on its diagonal, and zeros everywhere else. We place the correct parameters on the gain diagonal to scale the adaptation gains correctly, since the parameters range in magnitude from 10^{-1} to 10^4. For $\boldsymbol{\Gamma}_r$, we use $10^{-3} * \mathbf{I}$. Finally, we let $\lambda = 1.5$, and

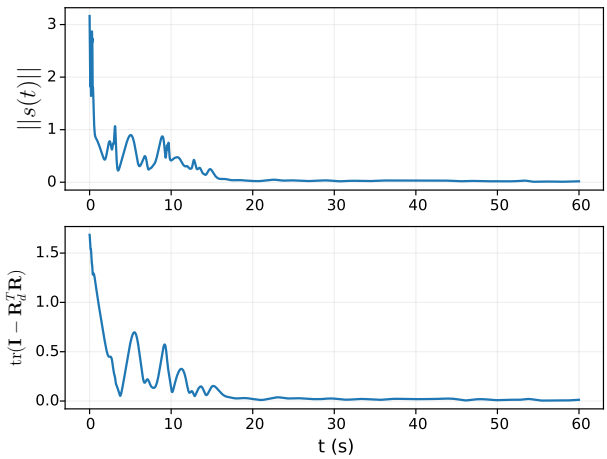
parameter	value
m	1.89×10^4 kg
I_{xx}	1.54×10^4 kg m^2
I_{yy}	1.54×10^4 kg m^2
I_{zz}	2.37×10^3 kg m^2
$\mathbf{r}_p$	$[0, 0, 1.5]$ m
$\mathbf{r}_1$	$[0, 0, 1.5]$ m
$\mathbf{r}_2$	$[0, 0, -1.5]$ m
$\mathbf{r}_3$	$[0.5, 0, 0]$ m
$\mathbf{r}_4$	$[-0.5, 0, 0]$ m
$\mathbf{r}_5$	$[0, 0.5, 0]$ m
$\mathbf{r}_5$	$[0, -0.5, 0]$ m

TABLE I. Parameter Values for Simulation in $SE(3)$

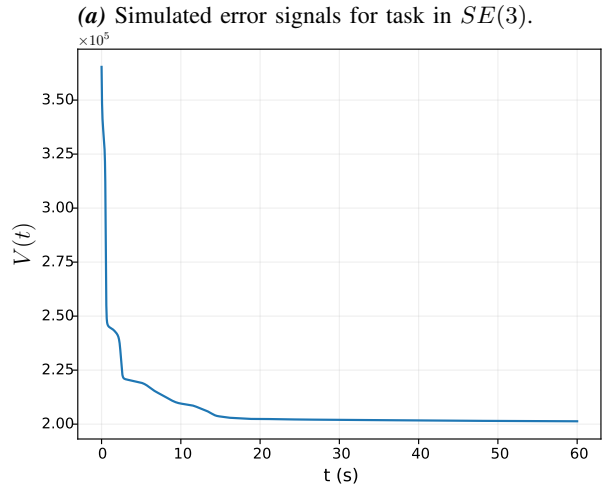
$\mathbf{K}_D = \text{diag}(\mathbf{K}_F, \mathbf{K}_M)$, where $\mathbf{K}_F = 5 \times 10^4 * \mathbf{I}$, and $\mathbf{K}_M = 5 \times 10^3 * \mathbf{I}$. The initial parameter estimates $\widehat{\mathbf{o}}_i$ and $\widehat{\mathbf{r}}_i$ were drawn from zero-mean multivariate Gaussian distributions with variances of $\Sigma_o = \mathbf{I}$ and $\Sigma_r = 2\mathbf{I}$, respectively. Source code of the simulation is publicly available at https://github.com/pculbertson/hamilton_ac.

Figure 2 plots the error signals and Lyapunov function value over the course of the simulation. We can see that the sliding variable is able to quickly reach the sliding surface $\mathbf{s} = 0$. Further, once on the sliding surface, we notice the rotation error metric also converges to zero, implying $\mathbf{R} \to \mathbf{R}_d$, as expected. We can further see that the Lyapunov function $V(t)$ is strictly decreasing (implying $\dot{V}(t) \leq 0$), and converges to a constant, non-zero value. This implies that the parameter errors $\widetilde{\mathbf{o}}_i, \widetilde{\mathbf{r}}_i$ do not converge to zero, which matches intuition due to the lack of a persistently exciting signal, as shown in Section VI-D.

We further compared the controller's performance against some simple baselines; namely, we studied the controller's performance when agents only adapt to the common object parameters $\mathbf{o}_i$, as studied in [27], [28], and in the case of no adaptation, which reduces to the case of a simple PD controller, since all parameter values are initialized to zero. The error signals for both cases are plotted in Figure 3. In the first case ($\mathbf{\Gamma}_r = \mathbf{0}$), we can see that that controller performance greatly suffers due to the inaccurate moment arm estimates. Specifically, in this case, the sliding variable $\mathbf{s} \not\to \mathbf{0}$; in fact, even over a very long time horizon, we found the sliding error never vanishes. We also found that by simply doubling the variance of the distribution from which the moment arms $\widehat{\mathbf{r}}_i$ were sampled, i.e., sampling $\widehat{\mathbf{r}}_i \sim \mathcal{N}(\mathbf{0}, 2\sqrt{2}\mathbf{I})$, would cause the controller could diverge. This behavior is as expected, since when the moment arm estimates are poor, there is a large, unmodelled coupling between the rotational and translational dynamics for which controller cannot compensate. In the second case, we plot the performance which results from only using the PD term in the control law (12), to demonstrate the value of the proposed adaptation routine. We can clearly see that the PD term alone is unable to send the sliding error $\mathbf{s}$ to zero; thus the proposed adaptation is essential to ensure that the tracking error vanishes.



(a) Simulated error signals for task in $SE(3)$.



(b) Simulated Lyapunov function $V(t)$ for $SE(3)$ control.

Fig. 2. Simulation results for a group of $N = 6$ agents controlling a body in $SE(3)$. ***(a)***: Simulated values of the sliding variable norm $||\mathbf{s}(t)||$ and rotation error. We notice these signals converge to zero, with the rotation error converging once the system remains on the sliding surface. ***(b)***: Simulated value of the Lyapunov function $V(t)$. We notice it is non-increasing (i.e. $\dot{V}(t) \leq 0$), and converges to a constant, non-zero value, since the parameter errors do not vanish.

We were also interested in how the controller performs in the case of single-agent failure (i.e. if one or more agents are deactivated at some time t_d during the manipulation task). We note that this is an instantaneous shift in the system parameters (since new α_i must be chosen such that the constants still sum to one). This is, in turn, equivalent to restarting the controller with new

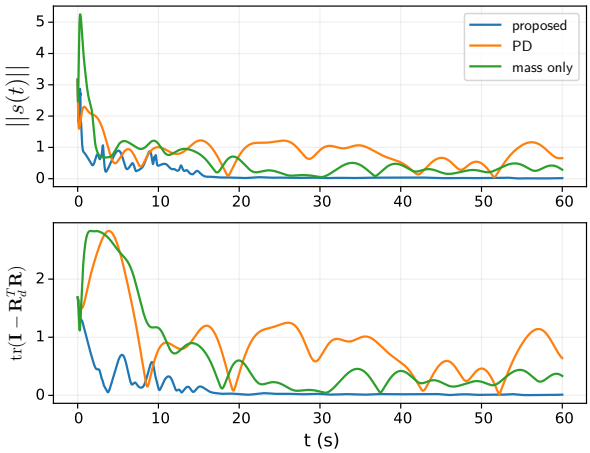


Fig. 3. Simulated baseline comparisons for the $SE(3)$ manipulation task. In green, we plot the error signals in the case where the agents estimate and adapt to only the object parameters $\widehat{\mathbf{o}}_i$, as in [27], [28]. In this case, neither the sliding variable $\mathbf{s}$ nor the rotation error converges to zero, even if simulated over a long horizon. The controller may also diverge for some initial conditions. In orange, the agents implement a PD controller by setting their parameter estimates $\widehat{\mathbf{o}}_i, \widehat{\mathbf{o}}_i$ to zero for all time. We note the error does not converge to zero, and tracking performance is generally poor.

system parameters $\mathbf{o}_i$, and a different initial condition; thus the boundedness and convergence results still hold. Figure 4 plots the system behavior in the case that three agents (half the manipulation team) are deactivated at $t_d = 30$ s. As expected, when the agents are dropped, we see a discontinuity in the Lyapunov function $V(t)$, due to the parameter change, and a slight jump in the sliding error $\mathbf{s}$. However, the remaining three agents are still able to regain tracking, as expected.

Finally, as shown recently in [42], the parameter error terms in the Lyapunov function (9), $\widetilde{\mathbf{a}}_i^T\mathbf{\Gamma}_i^{-1}\widetilde{\mathbf{a}}_i$ can be generalized by instead using the Bregman divergence $d_\psi(\mathbf{\Gamma}_i\mathbf{a}_i \;\|\; \mathbf{\Gamma}_i\widehat{\mathbf{a}}_i)$, where, for some strictly convex function ψ, we write the Bregman divergence as

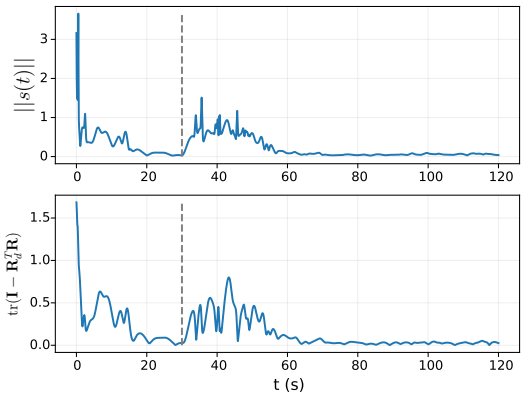
$$d_\psi(\mathbf{x} \,\|\, \mathbf{y}) = \psi(\mathbf{y}) - \psi(\mathbf{x}) - (\mathbf{y} - \mathbf{x})^T\nabla\psi(\mathbf{x}).$$

If we use the adaptation law

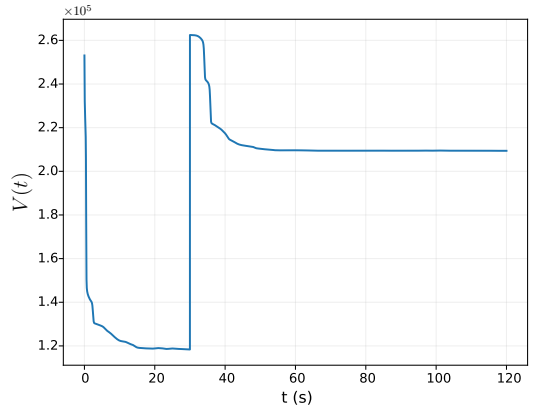
$$\dot{\widehat{\mathbf{a}}}_i = -\mathbf{\Gamma}_i^{-1}(\nabla^2\psi(\mathbf{\Gamma}_i\hat{\mathbf{a}}))^{-1}\mathbf{\Gamma}_i^{-1}\mathbf{Y}_i^T\mathbf{s},$$

we can easily reproduce the boundedness and convergence results shown previously. In fact, one can show the usual quadratic penalty is simply a Bregman divergence for $\psi(\mathbf{x}) = \frac{1}{2}||\mathbf{\Gamma}^{-\frac{1}{2}}\mathbf{x}||_2^2$.

By replacing the quadratic term with the more general Bregman divergence, we now have an adaptation law which uses a "natural gradient" with respect to the Hessian metric of the convex function ψ. The idea of using a natural gradient for adaptation, in order to impose underlying Riemannian constraints on the parameters, was introduced in [42], and generalized by [43]. In



(a) Simulated error signals, 3 agents shut off at $t = 30$.



(b) Simulated Lyapunov function during agent shutoff.

Fig. 4. Simulation results for $SE(3)$ task, where 3 of the $N = 6$ agents are deactivated at time $t = 30$. ***(a)***: Simulated values of the sliding variable norm $||\mathbf{s}(t)||$ and rotation error. The "drop time" is shown with a dashed grey line. After the drop, the remaining agents return to tracking the reference trajectory. ***(b)***: Simulated value of the Lyapunov function $V(t)$. We note the function is non-increasing, except for the discontinuity at $t = 30$, which occurs due to the instantaneous switch of parameter values.

[44], the authors explore this idea further, and show that the choice of ψ actually allows different forms of regularization to be imposed on the parameter estimates. This is especially important due to the inherent overparameterization of our system.

As a cursory study of this behavior, we substituted the usual quadratic penalty for a Bregman divergence with $\psi(\mathbf{x}) = ||\mathbf{x}||_1$, the ℓ_1 norm. In [44], the authors show the ℓ_1 norm leads to sparse solutions, meaning the parameter estimates converge to vectors which can achieve tracking with few non-zero values. Since our system is highly over-parameterized, we were interested in comparing this sparse solution with those achieved using the typical Euclidean geometry.

For this experiment, we increased the number of agents to $N = 100$ to exaggerate the overparameterization of the system. While the tracking performance and transient error were quite similar for both regularization strategies, they give rise to quite different estimates of the object parameters. Figure 6 plots histograms of the estimated parameters under both the ℓ_1 norm and ℓ_2 norm divergences. Here the parameters are scaled so all values are on the same order. We can indeed see the ℓ_1 solution is much sparser, with many values equal to zero, and with more outliers than the ℓ_2 case. We can further see the ℓ_2 norm results in a more disperse parameter distribution, as expected.

Figure 6 shows a visualization of the estimated parameters. We can again see that the ℓ_1 solution is quite sparse, with many entries equal to zero, in addition to the few outliers which are have relatively large values. In contrast, the ℓ_2 solution shows a "banded" structure that implies agreement between agents, where all agents arrive at parameter estimates that are somewhat small in magnitude, and roughly equal across agents.

Again, while these experiments are quite cursory, they demonstrate that due to the overparameterization of the system, we can design various forms of regularization to express preferences about which parameter estimates, among those that achieve tracking, are desirable. The question of how various regularization terms affect the system's robustness to noise, or the overall control effort, could prove interesting directions for future research.

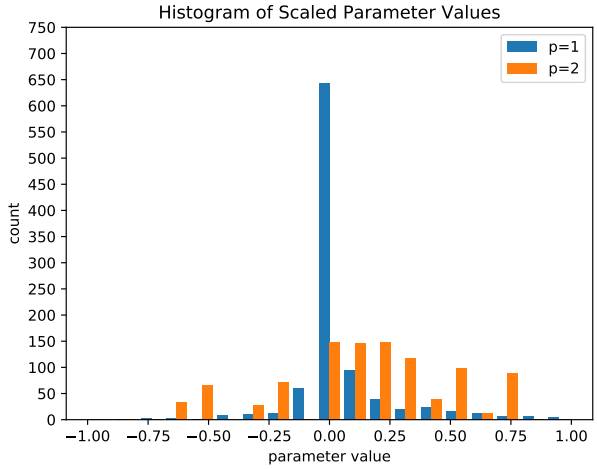


Fig. 5. Histogram of estimated parameter values for ℓ_1 and ℓ_2 regularization. When using the ℓ_1 norm (blue), we see the resulting parameter estimate is quite sparse, with a large spike centered at 0. When using the ℓ_2 norm (orange), we instead see the distribution of parameter estimates is less peaked, but takes on fewer large values.

VIII. Experimental Results

We further studied the performance of the proposed controller experimentally, using a group of $N = 6$ ground robots performing a manipulation task in the plane. Figure 7 shows one of the robots used in this experiment, a Ouijabot [45], which is a holonomic ground robot designed specifically for collaborative manipulation. The robots were equipped with current sensors on each motor, which allows them to estimate and control the wrench they are applying to a payload. Specifically, the robots used a low-level PID controller to generate motor torques, based on the desired wrench set by the higher-level adaptive controller, with feedback from the current sensors. They are also equipped with a Raspberry Pi 3B+, which allows them to perform onboard computation, as well as receive state measurements via ROS. The robots did not use any other communication capabilities.

Figure 8 shows the payload which the robots manipulated, which was an asymmetric wood piece that weighed roughly 5.5 kg, which was too heavy for any individual agent to move. The robots were rigidly attached to the piece, which remained roughly 4 cm from the ground. As seen in the figure, the measurement point $\mathbf{r}_p$ coincided with one of the robots, and was located quite far from the center of mass, resulting in long moment arms $\mathbf{r}_i$ as well as non-negligible coupling between the rotational and translational dynamics. The robots received position measurements from this point using Optitrack, a motion capture system; the velocity measurements were computed by numerically differentiating the position measurements. All signals were passed through a first-order filter to reduce the effects of measurement noise.

We note this experiment introduced considerable sources of modelling error which were not parameterized by the adaptive controller. Specifically, the wrenches computed by the high-level controller were only setpoints for a low-level PID controller. Further, while the control law included terms for both viscous and Coulomb friction, as outlined in Section VI-E, this of course did not capture the complex frictional relationship between the robot wheels and foam floor of the experimental space, which depended on wheel speed, direction, etc. Finally, all control signals were subject to time delays, discretization error, and measurement noise. We found our controllers (which were implemented as ROS nodes) ran at a frequency of 20 Hz.

Figure 9 plots the results of the experimental study. The desired trajectory $\mathbf{q}_d$ of the measurement point corresponded to a "figure-8" in the $x - y$ dimension, with the desired rotation θ_d oscillating between $-\frac{\pi}{4}$ and $\frac{\pi}{4}$ at successive waypoints. The experiment included 25 trials of the manipulation task, with all initial parameter estimates set to zero. Figure 9a plots the sliding error

Fig. 6. Visualization of estimated parameters for each regularization term. **Top:** Visualization of the estimated parameters when using the Bregman divergence for the ℓ_1 norm. In this case, we can see this regularization does indeed promote sparsity, with many entries being equal or close to zero, with a few outliers taking on larger absolute values. **Bottom:** Visualization of the estimated parameters when using the ℓ_2 norm. In this case, the parameters take on much smaller values, but are roughly equal across all agents.

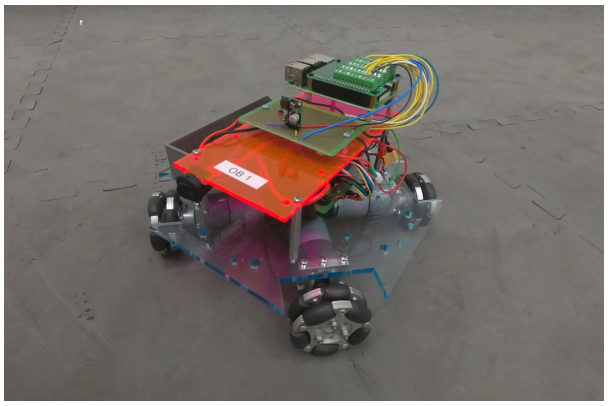


Fig. 7. Photo of a Ouijabot, the holonomic ground robot used to conduct the experiments presented in this work. The robot is equipped with current sensors on its motors, which allows it to close a feedback loop to provide a desired force and torque to an external payload.

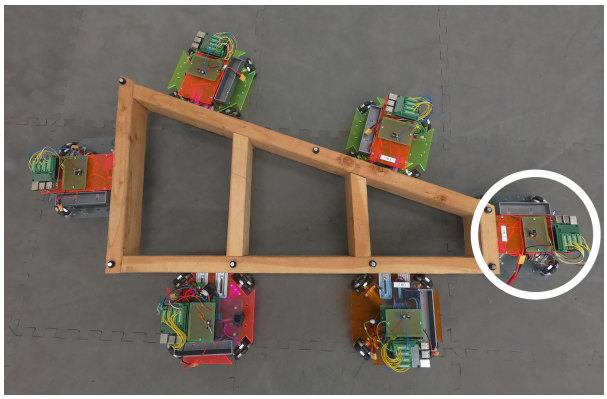


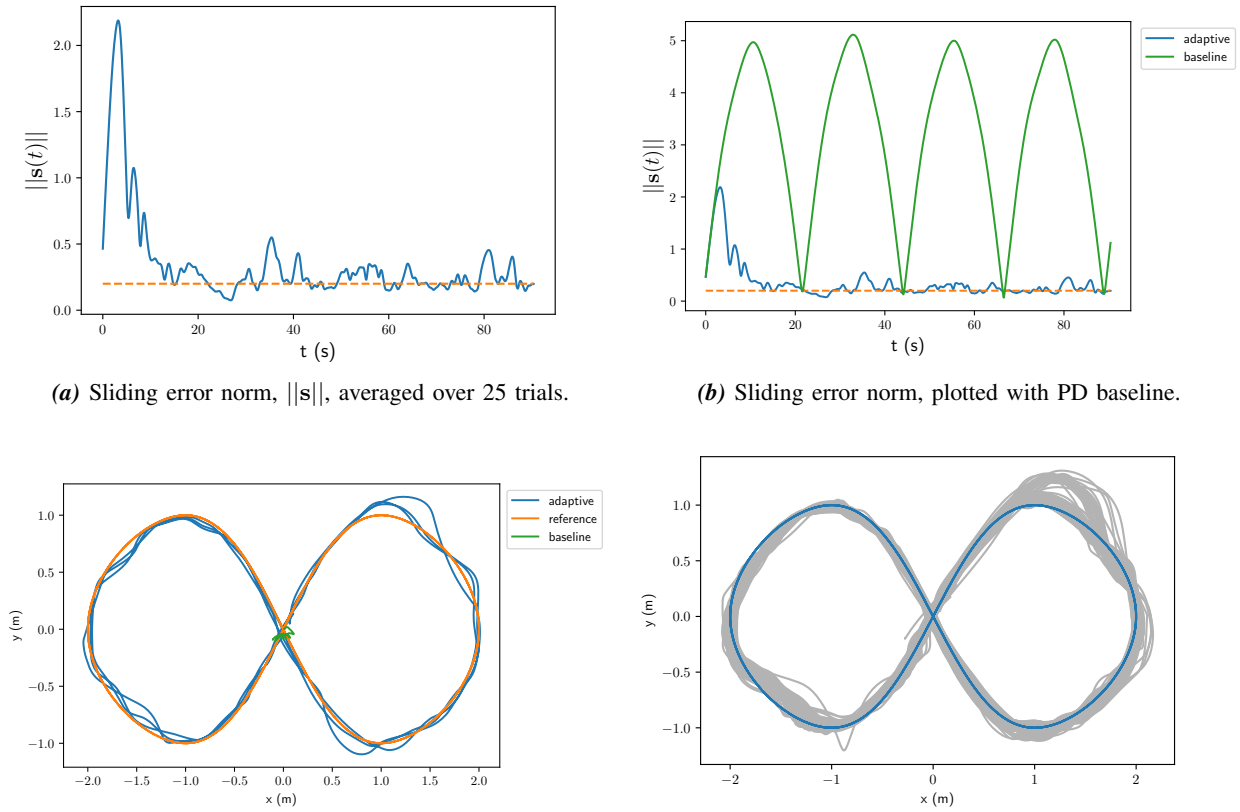
Fig. 8. Experimental setup for planar manipulation task. A team of $N = 6$ ground robots manipulate a shared wood payload. The measurement point for the payload coincides with one of the agents, circled in white, and is located far from the geometric center of the body, and its center of mass.

norm, averaged over the 25 trials, with the deadband of 0.2, under which adaptation was stopped, plotted as a dotted line. We can see that the controller converges to be within the deadband in about 45 seconds. Figure 9c shows the ground track of a representative trial, with the reference trajectory plotted in blue, and the response of the underlying PD controller plotted in green. The initial state of the robot is the center of the "figure-8," from which it moves up and to the left into the first turn of the desired trajectory. A single experimental run consisted of three cycles of the reference signal, which ran over roughly 100 s. We purposely used quite low PD gains in order to demonstrate the effectiveness of the adaptation routine; the PD control alone gives essentially no response to the reference trajectory.

We can see the robots initially overshoot the first turn by a large amount, before achieving improved tracking. We further note that even after the robots have mostly converged, they still leave the deadband for brief moments, which correspond to the turns of the "figure-8," which proved more difficult to track. These errors would likely be eliminated with more adaptation. Figure 9d plots the ground track of all 25 trials, with the reference trajectory shown in blue. We can see the controller's performance is consistent across trials.

IX. Concluding Remarks

This work has focused on the problem of collaborative manipulation without knowledge of payload parameters or explicit communication between agents. To this end, we proposed a sliding model control architecture on



(a) Sliding error norm, $||\mathbf{s}||$, averaged over 25 trials.

(b) Sliding error norm, plotted with PD baseline.

(c) Ground tracks for one experimental trial.

(d) Ground tracks for all trials (gray), with reference (blue).

Fig. 9. Experimental results for a group of $N = 6$ ground robots performing a planar manipulation task. ***(a)***: Measured value of the sliding error, averaged across 25 trials. ***(b)*** Sliding error, plotted with the values measured when only using the PD term of the control law. ***(c)***: Overhead view of the measurement point trajectory for one trial. ***(d)***: Overhead plot of all ground tracks across the 25 trials, with the reference trajectory shown in blue.

$SE(3)$, together with a distributed adaptation law, which enabled the agents to gradually adapt to the unknown object parameters to track a desired trajectory. We further provided a Lyapunov-like proof of the asymptotic convergence of the system to the reference trajectory under our proposed controller, and studied its performance for practical manipulation tasks using both numerical simulations and hardware experiments.

There exist numerous directions for future work. An important open question is how to incorporate actuation constraints, or underactuation, into the proposed control framework; in this work we assume each agent can create any desired force and torque on the object at its attachment point. Another interesting direction is to understand more clearly the connections between this work, and other distributed control architectures such as quorum sensing or the control of networked systems. As mentioned previously, studying the effect of different regularization strategies, or of using higher-order adaptation laws, on the overall system performance could also prove valuable. Further, while the proposed controller is robust to slow changes in the object's mass and geometric properties, another interesting direction would be to study how and if the proposed architecture can be extended to the manipulation of articulated or flexible bodies. Finally, while we assume the agents travel through free space, we are quite interested in enabling distributed manipulation in the presence of obstacles, and studying how agents can coordinate themselves through object motion to avoid obstacles they cannot directly observe.

References

[1] O. Khatib, K. Yokoi, O. Brock, K. Chang, and A. Casal. Robots in human environments: Basic autonomous capabilities. *The International Journal of Robotics Research*, 18(7):684–696, 1999.

[2] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, A. Casal, and A. Baader. Force strategies for cooperative tasks in multiple mobile manipulation systems. In Georges Giralt and Gerhard Hirzinger, editors, *Robotics Research*, pages 333–342, London, 1996. Springer London.

[3] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, and A. Casal. Coordination and decentralized cooperation of multiple mobile manipulators. *Journal of Robotic Systems*, 13(11):755–764, November 1996.

[4] J. Fink, M. A. Hsieh, and V. Kumar. Multi-robot manipulation via caging in environments with obstacles. In *2008 IEEE International Conference on Robotics and Automation*, pages 1471–1476, May 2008.

[5] D. Rus, B. Donald, and J. Jennings. Moving furniture with teams of autonomous robots. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*. IEEE Comput. Soc. Press, 1995.

[6] C. Alford and S. Belyeu. Coordinated control of two robot arms. In *Proceedings. 1984 IEEE International Conference on Robotics and Automation*, volume 1, pages 468–473, March 1984.

[7] Oussama Khatib. Object manipulation in a multi-effector robot system. In *Proceedings of the 4th International Symposium on Robotics Research*, pages 137–144, Cambridge, MA, USA, 1988. MIT Press.

[8] John T. Wen and Kenneth Kreutz-Delgado. Motion and force control of multiple robotic manipulators. *Automatica*, 28(4):729 – 743, 1992.

[9] R. C. Bonitz and T. C. Hsia. Internal force-based impedance control for cooperating manipulators. *IEEE Transactions on Robotics and Automation*, 12(1):78–89, Feb 1996.

[10] F. Caccavale, P. Chiacchio, A. Marino, and L. Villani. Six-dof impedance control of dual-arm cooperative manipulators. *IEEE/ASME Transactions on Mechatronics*, 13(5):576–586, Oct 2008.

[11] S. Erhart, D. Sieber, and S. Hirche. An impedance-based control architecture for multi-robot cooperative dual-arm mobile manipulation. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 315–322, Nov 2013.

[12] S. Erhart and S. Hirche. Internal force analysis and load distribution for cooperative multi-robot manipulation. *IEEE Transactions on Robotics*, 31(5):1238–1243, Oct 2015.

[13] Karl Böhringer, Russell Brown, Bruce Donald, Jim Jennings, and Daniela Rus. Distributed robotic manipulation: Experiments in minimalism. In Oussama Khatib and J. Kenneth Salisbury, editors, *Experimental Robotics IV*, pages 11–25, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.

[14] Peng Song and V. Kumar. A potential field based approach to multi-robot manipulation. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, volume 2, pages 1217–1222 vol.2, May 2002.

[15] Siamak G. Faal, Shadi Tasdighi Kalat, and Cagdas D. Onal. Towards collective manipulation without inter-agent communication. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, SAC '16, pages 275–280, New York, NY, USA, 2016. Association for Computing Machinery.

[16] Daniel Mellinger, Michael Shomin, Nathan Michael, and Vijay Kumar. *Cooperative Grasping and Transport Using Multiple Quadrotors*, pages 545–558. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[17] Koushil Sreenath and Vijay Kumar. Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots. In *Robotics: Science and Systems IX*. Robotics: Science and Systems Foundation, June 2013.

[18] Sarah Tang, Koushil Sreenath, and Vijay Kumar. Multi-robot trajectory generation for an aerial payload transport system. In Nancy M. Amato, Greg Hager, Shawna Thomas, and Miguel Torres-Torriti, editors, *Robotics Research*, pages 1055–1071, Cham, 2020. Springer International Publishing.

[19] Nathan Michael, Jonathan Fink, and Vijay Kumar. Cooperative manipulation and transportation with aerial robots. *Autonomous Robots*, 30(1):73–86, Jan 2011.

[20] B. E. Jackson, T. A. Howell, K. Shah, M. Schwager, and Z. Manchester. Scalable cooperative transport of cable-suspended loads with uavs using distributed trajectory optimization. *IEEE Robotics and Automation Letters*, 5(2):3368–3374, 2020.

[21] Zijian Wang and Mac Schwager. Force-amplifying n-robot transport system (force-ants) for cooperative planar manipulation without communication. *The International Journal of Robotics Research*, 35(13):1564–1586, 2016.

[22] Z. Wang, S. Singh, M. Pavone, and M. Schwager. Cooperative object transport in 3d with multiple quadrotors using no peer communication. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1064–1071, May 2018.

[23] A. Franchi, A. Petitti, and A. Rizzo. Decentralized parameter estimation and observation for cooperative mobile manipulation of an unknown load using noisy measurements. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5517–5522, May 2015.

[24] A. Petitti, A. Franchi, D. Di Paola, and A. Rizzo. Decentralized motion control for cooperative manipulation with a team of networked mobile manipulators. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 441–446, May 2016.

[25] Yan-Ru Hu, Andrew A. Goldenberg, and Chin Zhou. Motion and force control of coordinated robots during constrained motion tasks. *The International Journal of Robotics Research*, 14(4):351–365, 1995.

[26] Z. Li, S. S. Ge, and Z. Wang. Robust adaptive control of coordinated multiple mobile manipulators. In *2007 IEEE International Conference on Control Applications*, pages 71–76, Oct 2007.

[27] Yun-Hui Liu and Suguru Arimoto. Decentralized adaptive and nonadaptive position/force controllers for redundant manipulators in cooperations. *The International Journal of Robotics Research*, 17(3):232–247, 1998.

[28] Christos K. Verginis, Matteo Mastellaro, and Dimos V. Dimarogonas. Robust quaternion-based cooperative manipulation without force/torque information**this work was supported by the h2020 erc starting grand bu-cophsys, the swedish research council (vr), the knut och alice wallenberg foundation and the european union's horizon 2020 research and innovation programme under the grant agreement no. 644128 (aeroworks). *IFAC-PapersOnLine*, 50(1):1754 – 1759, 2017. 20th IFAC World Congress.

[29] Domenico Prattichizzo and Jeffrey C. Trinkle. *Grasping*, pages 671–700. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[30] P. Culbertson and M. Schwager. Decentralized adaptive control for collaborative manipulation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 278–285, May 2018.

[31] Jean-Jacques E. Slotine and Weiping Li. On the adaptive control of robot manipulators. *The International Journal of Robotics Research*, 6(3):49–59, 1987.

[32] Nicolas Tabareau, Jean-Jacques Slotine, and Quang-Cuong Pham. How synchronization protects from noise. *PLoS Computational Biology*, 6(1):e1000637, January 2010.

[33] Giovanni Russo and Jean Jacques E. Slotine. Global convergence of quorum-sensing networks. *Phys. Rev. E*, 82:041919, Oct 2010.

[34] Christopher M. Waters and Bonnie L. Bassler. Quorum sensing: Cell-to-cell communication in bacteria. *Annual Review of Cell and Developmental Biology*, 21(1):319–346, 2005. PMID: 16212498.

[35] S. P. Bhat and D. S. Bernstein. A topological obstruction to global asymptotic stabilization of rotational motion and the unwinding phenomenon. In *Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No.98CH36207)*, volume 5, pages 2785–2789 vol.5, 1998.

[36] Richard M. Murray, Zexiang Li, and S. Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, December 2017.

[37] Jean-Jacques E. Slotine and Weiping Li. *Applied Nonlinear Control*. Prentice-Hall, Inc., 1991.

[38] G. C. Gómez Cortés, F. Castaños, and J. Dávila. Sliding motions on so(3), sliding subgroups. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 6953–6958, 2019.

[39] Jean-Jacques E. Slotine and Weiping Li. Theoretical issues in adaptive manipulator control. In *Proceedings of the 5th Yale Workshop on Applied Adaptive Systems Theory*, 1987.

[40] P. M. Wensing and J. Slotine. Cooperative adaptive control for cloud-based robotics. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6401–6408, 2018.

[41] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Springer New York, 1980.

[42] T. Lee, J. Kwon, and F. C. Park. A natural adaptive control law for robot manipulators. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9, 2018.

[43] Patrick M. Wensing and Jean-Jacques E. Slotine. Beyond convexity – contraction and global convergence of gradient descent, 2018.

[44] Nicholas M. Boffi and Jean-Jacques E. Slotine. Higher-order algorithms and implicit regularization for nonlinearly parameterized adaptive control, 2019.

[45] Zijian Wang, Guang Yang, Xuanshuo Su, and Mac Schwager. OuijaBots: Omnidirectional robots for cooperative object transport with rotation control using no communication. In *Distributed Autonomous Robotic Systems*, pages 117–131. Springer International Publishing, 2018.

[46] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

APPENDIX

A. Proofs of Matrix Properties

We can now discuss properties of $\mathbf{H}, \mathbf{C}$, and their derivatives. We first show the positive definiteness of $\mathbf{H}$. Physically, since the kinetic energy of the system can be expressed as

$$K = \frac{1}{2}\dot{\mathbf{q}}^T\mathbf{H}(\mathbf{q})\dot{\mathbf{q}},$$

we must have $\mathbf{H}(\mathbf{q}) \succ 0$, otherwise there exists $\dot{\mathbf{q}} \neq \mathbf{0}$ with $K = 0$. For a block matrix of the form

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix},$$

using the properties of the Schur complement [46], we can say $\mathbf{M} \succ 0$ if and only if $\mathbf{A} \succ 0$ and $\mathbf{C} - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B} \succ 0$.

Thus, since $m\mathbf{I} \succ 0$ by inspection, $\mathbf{H} \succ 0$ if and only if $\mathcal{S} = \mathbf{R}\mathbf{I}_p\mathbf{R}^T + m(\mathbf{R}\mathbf{r}_p)^\times(\mathbf{R}\mathbf{r}_p)^\times \succ 0$.

We further use the fact that

$$\mathbf{v}^\times\mathbf{v}^\times = \mathbf{v}\mathbf{v}^T - (\mathbf{v}^T\mathbf{v})\mathbf{I},$$

to write

$$\begin{aligned}\mathcal{S} &= \mathbf{R}\left(\mathbf{I}_{cm} \pm m(\mathbf{r}_p^T\mathbf{r}_p - \mathbf{r}_p\mathbf{r}_p^T)\right)\mathbf{R}^T, \\ &= \mathbf{R}\mathbf{I}_{cm}\mathbf{R}^T,\end{aligned}$$

and thus $\mathcal{S} \succ 0$, since $\mathbf{I}_{cm} \succ 0$ for bodies of finite size. Thus, we conclude $\mathbf{H} \succ 0$.

We can further show $\dot{\mathbf{H}} - 2\mathbf{C}$ is skew-symmetric, which can be understood as a matrix expression of conservation of energy. Specifically, using the principle of conservation of energy, we can write

$$\dot{\mathbf{q}}^T(\boldsymbol{\tau} - \mathbf{g}) = \frac{1}{2}\frac{d}{dt}\left[\dot{\mathbf{q}}^T\mathbf{H}(\mathbf{q})\dot{\mathbf{q}}\right].$$

Differentiating the right side yields

$$\dot{\mathbf{q}}^T(\boldsymbol{\tau} - \mathbf{g}) = \dot{\mathbf{q}}^T\mathbf{H}\ddot{\mathbf{q}} + \tfrac{1}{2}\dot{\mathbf{q}}^T\dot{\mathbf{H}}\dot{\mathbf{q}},$$

which, rearranging, results in

$$\dot{\mathbf{q}}^T(\dot{\mathbf{H}} - 2\mathbf{C})\dot{\mathbf{q}} = \mathbf{0},$$

which must hold true for all $\dot{\mathbf{q}}$. While the matrix $\mathbf{C}$ is non-unique, there exist choices of $\mathbf{C}$ where $\dot{\mathbf{H}} - 2\mathbf{C}$ is skew-symmetric, which is indeed the case for this system. Thus, we can view this matrix property as an expression of conservation of energy. We now demonstrate this property for our choice of $\mathbf{C}$.

$\dot{\mathbf{H}}$ is given by

$$\dot{\mathbf{H}} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{A} - \mathbf{A}^T \\ \mathbf{A}^T - \mathbf{A} & \boldsymbol{\omega}^\times\mathbf{R}\mathbf{I}_p\mathbf{R}^T - \mathbf{R}\mathbf{I}_p\mathbf{R}\boldsymbol{\omega}^\times \end{bmatrix},$$

where $\mathbf{A} = m\boldsymbol{\omega}^\times(\mathbf{R}\mathbf{r}_p)^\times$, and using the fact $\dot{\mathbf{R}} = \boldsymbol{\omega}^\times\mathbf{R}$. Further,

$$\dot{\mathbf{H}} - 2\mathbf{C} = \begin{bmatrix} \mathbf{0}_3 & -\mathbf{B} \\ \mathbf{B} & \mathbf{D} \end{bmatrix},$$

where $\mathbf{B} = \mathbf{B}^T = \mathbf{A} + \mathbf{A}^T$ and $\mathbf{D} = -\boldsymbol{\omega}^\times\mathbf{R}\mathbf{I}_p\mathbf{R}^T - \mathbf{R}\mathbf{I}_p\mathbf{R}\boldsymbol{\omega}^\times - m((\mathbf{R}\mathbf{r}_p)^\times\dot{\mathbf{x}})^\times$. We can see $\dot{\mathbf{H}} - 2\mathbf{C}$ is skew-symmetric, since $\mathbf{D}$ is skew-symmetric, and the off-diagonal blocks are symmetric and of opposite sign.