

Decentralized Adaptive Control for Collaborative Manipulation

Preston Culbertson¹ and Mac Schwager²

Abstract—This paper presents a design for a decentralized adaptive controller that allows a team of agents to manipulate a common payload in \mathbb{R}^2 or \mathbb{R}^3 . The controller requires no communication between agents and requires no *a priori* knowledge of agent positions or payload properties. The agents can control the payload to track a reference trajectory in linear and angular velocity with center-of-mass measurements, in angular velocity using only local measurements and a common frame, and can stabilize its rotation with only local measurements. The controller is designed via a Lyapunov-style analysis and has proven stability and convergence. The controller is validated in simulation and experimentally.

I. INTRODUCTION

Collaborative manipulation remains one of the most important problems in multi-agent systems. Using teams of robots to manipulate large or heavy payloads promises many advantages over single-agent manipulators, including flexibility, scalability, and robustness to individual agent failures. Applications for these systems include construction, manufacturing and assembly, search and rescue, and debris removal. However, current strategies make restrictive assumptions which prevent these systems from being used in the field.

A. Contribution

This paper presents a design for a decentralized adaptive controller that allows a team of agents to manipulate a common payload in \mathbb{R}^2 or \mathbb{R}^3 . The controller has proven stability and requires no communication between agents and requires no *a priori* knowledge of agent positions or payload properties.

In this paper, we solve three collaborative manipulation problems of varying difficulty, depending on the information available to the agents. Firstly, we solve the problem of general manipulation, where the payload's angular and linear velocities are controlled. In the second problem, we aim to track only a desired angular velocity for the body. We find this can be achieved using only local measurements from each agent, assuming they share a common body-fixed frame. Further, without a common frame, the agents can still achieve stabilization of the body's angular rates. These strategies apply to agents in \mathbb{R}^2 and \mathbb{R}^3 , and are robust to the addition or removal of agents.

These results are consequences of a novel decentralized adaptive control design strategy, which is introduced in this paper for a class of nonlinear dynamical systems. The

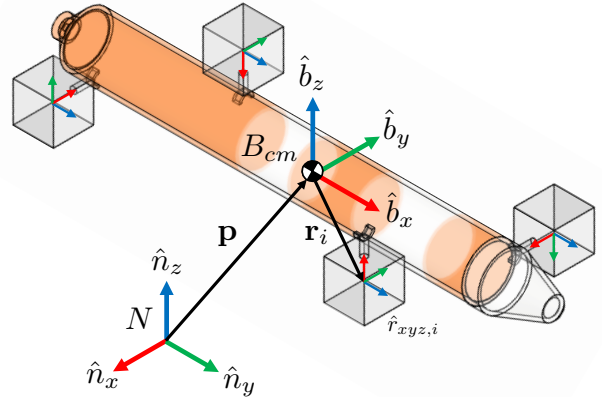


Fig. 1: Four agents manipulate a common payload. The agents have no prior knowledge of the object's mass or frictional properties, and cannot communicate with each other. Using the proposed decentralized adaptive controllers, they can control the body's linear and angular velocities to track a desired trajectory.

controller is designed via a Lyapunov-style analysis, and has proven convergence and stability properties.

We experimentally demonstrate a team of omni-directional ground robots [1] manipulating a common payload in \mathbb{R}^2 , and present a simulation of a team manipulating a payload in \mathbb{R}^3 using Gazebo [2], an open-source dynamics engine.

B. Prior Work

Collaborative manipulation has been extensively studied, beginning with a set of protocols for pushing rigid objects in [3]. Various decentralized manipulation strategies have been proposed, including force sensing [4], potential fields [5], caging [6], consensus [7], and pseudo-inverse allocation [8]. In [9] the authors demonstrate automated transport and assembly of furniture with ground robots. In [10] the authors develop and demonstrate a strategy for manipulating flexible payloads. Further, various recent papers have focused on cooperative manipulation with aerial vehicles, using both cables [11] and rigid attachments [12], as well as path planning for cooperative lifting [13]. These algorithms, while successful, suffer from the significant drawback of requiring accurate payload knowledge, including the locations of each agent from the center of mass, as well as mass and frictional properties of the payload.

To this end, some authors have sought to develop strategies that estimate or adapt to unknown payload dynamics. In [14] and [15] the authors develop a decentralized strategy for estimating payload parameters, and using these estimates for manipulation. However, these algorithms make extensive use of networking between agents, and do not perform estimation and control simultaneously.

Previous work has also proposed applying adaptive control to manipulation tasks. In [16], the authors develop

This work was supported in part by NSF Grant IIS-1646921. We are grateful for this support.

¹P. Culbertson is with the Department of Mechanical Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: pculbertson@stanford.edu).

²M. Schwager is with the Department of Aeronautics and Astronautics, Stanford University (e-mail: schwager@stanford.edu).

a centralized adaptive controller to control the attitude of a quadrotor. In [17], the authors develop a decentralized adaptive controller that allows robotic arms to adapt to their own dynamics, but consider the payload properties to be known. Further, in [18], the authors develop a fully centralized adaptive controller for multiple manipulators moving an unknown payload.

Our work applies results from Model Reference Adaptive Control (MRAC) [19]. Further, our work is related to decentralized adaptive control, wherein adaptive controllers are designed for interconnected systems, as in [20], [21].

The remainder of the paper is structured as follows. In Section II, we formulate three collaborative manipulation tasks, and outline the assumptions required to accomplish them. In Section III, we formulate the dynamics of the manipulation problems. Section IV presents a decentralized adaptive controller design for a class of nonlinear dynamical systems that includes the dynamics described in Section III. In Section V, we assess our controller performance for a manipulation task in \mathbb{R}^3 using an open-source dynamics engine. In Section VI, we present the experimental results of our controller for a planar manipulation task using ground-based mobile robots.

II. PROBLEM STATEMENT

We consider a team of n robots, $R_i, i \in \{1, \dots, n\}$, rigidly attached to a rigid payload B , which has mass m and inertia matrix \mathbf{I} . Each agent has a body-fixed coordinate frame, $\hat{\mathbf{r}}_{xyz,i}$, and there also exists a body-fixed frame $\hat{\mathbf{b}}_{xyz}$ in B . Each agent is attached at position \mathbf{r}_i , measured from the object's center of mass, and is capable of applying a force \mathbf{F}_i and a torque \mathbf{T}_i to the object. The body is located at position \mathbf{p} , and has a velocity \mathbf{v}_{cm} measured at its center of mass, and angular velocity $\boldsymbol{\omega}$ measured in Newtonian frame N . The body is also subject to a frictional force \mathbf{F}_f (e.g. sliding friction or air resistance). A schematic of the system is shown in Figure 1.

Each agent has a low-level controller which can apply the desired forces \mathbf{F}_i and torques \mathbf{T}_i . The agents are also equipped with sensors to measure their linear velocity \mathbf{v}_i and angular velocity $\boldsymbol{\omega}_i$ in N . Each agent also has access to a reference signal and model trajectory, either via a broadcast from a ground station, or precomputed and stored onboard. The agents cannot communicate with each other or the ground station, and have no *a priori* knowledge of the payload's mass or frictional properties.

We now formulate three collaborative manipulation problems to be solved in this paper. The problems are ordered from most to least challenging, and each requires progressively less restrictive assumptions for its solution.

Problem 1 (Collaborative Manipulation). *Consider a team of n agents manipulating a rigid object B . Let $\mathbf{x} = [\mathbf{v}_{cm}^T \ \boldsymbol{\omega}^T]^T$ denote the object's appended linear and angular velocities. Design a decentralized controller such that, given a reference velocity $\mathbf{x}_m(t)$, the control law guarantees stability and asymptotic tracking.*

To solve this problem, we make the following assumptions:

Assumption 1 (Center-of-Mass Measurement). *Each agent has access to $\mathbf{x}(t)$, the linear and angular velocity of B measured at its center of mass.*

Assumption 2 (Common Reference Frame). *The agents share a common reference frame fixed in B (i.e. $\hat{\mathbf{b}}_{xyz}$), and know the orientation between their frame $\hat{\mathbf{r}}_{xyz,i}$ and this frame.*

While Assumption 1 appears restrictive, we discuss here multiple options for its removal.

One option is to place a sensor at the center of mass which broadcasts measurements to the agents. While this requires prior knowledge of B , an adaptive algorithm still allows for the solution of Problem 1 without measuring \mathbf{r}_i for each agent, which can be prohibitively expensive for large teams of agents.

In addition, Assumption 1 is satisfied by each agent knowing its location \mathbf{r}_i with respect to the center of mass. This is due to $\mathbf{w} = \mathbf{w}_i$, and \mathbf{v}_{cm} can be calculated using $\mathbf{v}_{cm} = \mathbf{v}_i + \boldsymbol{\omega}_i \times \mathbf{r}_i$. We seek to remove this assumption in future work by allowing measurements to be taken from any point on the body.

We also note that Assumption 2 is only mildly restrictive. Assuming each agent has a compass or star tracker onboard, it may use this device to determine its orientation with respect to the body frame.

Further, we can consider additional manipulation problems that allow us to remove Assumptions 1 and 2.

Problem 2 (Collaborative Rotation Control). *Consider a team of n agents manipulating a rigid object B . Design a decentralized controller such that, given a reference angular velocity $\boldsymbol{\omega}_m(t)$, the control law guarantees stability and asymptotic tracking.*

We note that solving this problem does not require Assumption 1, since angular velocity is equal at all points on a rigid body, and thus each agent may use its local measurement, $\boldsymbol{\omega}_i$ for control.

We can further restrict the problem to also remove Assumption 2.

Problem 3 (Collaborative Rotation Stabilization). *Consider a team of n agents manipulating a rigid object B . Design a decentralized control law that guarantees stability and drives the angular velocity to zero.*

While this problem is much more restricted than Problem 1, it forms an important class of manipulation problems which includes the stabilization of orbital debris for active removal as proposed in [22].

III. SYSTEM DYNAMICS

In this section, we formulate the dynamics of collaborative manipulation tasks in \mathbb{R}^2 and \mathbb{R}^3 as nonlinear dynamical systems. Throughout this discussion, we let $\frac{\mathcal{F}d\mathbf{v}}{dt}$ denote the time derivative of vector \mathbf{v} in the reference frame \mathcal{F} .

A. Planar Manipulation

We first consider a payload that is being manipulated in \mathbb{R}^2 by a team of agents. The linear dynamics are described

by $\mathbf{F} = m\mathbf{a}_{cm}$, where \mathbf{F} is the resultant of all forces on the body B , and \mathbf{a}_{cm} is the acceleration of the payload's center of mass in N . For planar manipulation, we consider both applied forces from the robots and frictional forces on the body, $\mathbf{F} = \sum_{i=1}^n \mathbf{F}_i + \mathbf{F}_f$, where we model the frictional force \mathbf{F}_f as

$$\mathbf{F}_f = -\mu_{0,\ell} \text{sgn} \mathbf{v}_{cm} - \mu_{1,\ell} \mathbf{v}_{cm},$$

where $\mu_{0,\ell}, \mu_{1,\ell}$ are zeroth-order (sliding) and first-order (viscous) frictional constants, and

$$\text{sgn}(x) = \begin{cases} +1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}.$$

Additionally, we can write the linear acceleration as

$$\mathbf{a}_{cm} = \frac{N d\mathbf{v}_{cm}}{dt} = (\dot{v}_x - \omega v_y) \hat{b}_x + (\dot{v}_y + \omega v_x) \hat{b}_y.$$

Further, the rotational dynamics of B are given by $\mathbf{M}^{B/B_{cm}} = J\boldsymbol{\alpha}$, where $\mathbf{M}^{B/B_{cm}}$ is the resultant of all moments on B about its center of mass, J is B 's moment of inertia about \hat{b}_z , and $\boldsymbol{\alpha}$ denotes B 's angular acceleration in N .

We consider both applied and frictional moments on B , $\mathbf{M}^{B/B_{cm}} = \mathbf{M}_f + \sum_{i=1}^n \mathbf{T}_i + \mathbf{r}_i \times \mathbf{F}_i$, where we model the frictional moment about the center of mass as

$$\mathbf{M}_f = (-\mu_{0,r} \text{sgn} \omega - \mu_{1,r} \omega) \hat{b}_z,$$

for frictional constants $\mu_{0,r}, \mu_{1,r}$.

Let $\mathbf{x} = \begin{bmatrix} v_x & v_y & \omega \end{bmatrix}^T$ denote the system state and $\mathbf{u}_i = \begin{bmatrix} F_{xi} & F_{yi} & T_i \end{bmatrix}^T$ denote the system input for agent i .

Using the equations above, we can write

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \sum_{i=1}^n \mathbf{B}_i(\mathbf{u}_i + \mathbf{L}_i \mathbf{f}(\mathbf{x})), \quad (1)$$

where

$$\mathbf{A} = \begin{bmatrix} -\frac{\mu_{1,\ell}}{m} & 0 & 0 \\ 0 & -\frac{\mu_{1,\ell}}{m} & 0 \\ 0 & 0 & -\frac{\mu_{1,r}}{m} \end{bmatrix}, \mathbf{B}_i = \begin{bmatrix} \frac{1}{m} & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ -\frac{r_{yi}}{J} & \frac{r_{xi}}{J} & \frac{1}{J} \end{bmatrix},$$

$$\mathbf{L}_i = \frac{1}{n} \mathbf{B}_i^{-1} \begin{bmatrix} \frac{\mu_{0,\ell}}{m} & 0 & 0 & 1 \\ 0 & \frac{\mu_{0,\ell}}{m} & 0 & -1 \\ 0 & 0 & \frac{\mu_{0,r}}{m} & 0 \end{bmatrix},$$

and $\mathbf{f}(\mathbf{x}) = \begin{bmatrix} \text{sgn}(v_x) & \text{sgn}(v_y) & \text{sgn}(\omega) & \omega v_x & \omega v_y \end{bmatrix}^T$.

B. General 3D Motion

We now consider a payload being manipulated in \mathbb{R}^3 . We now only consider a first-order (viscous) friction model,

$$\mathbf{F}_f = -\mu_\ell \mathbf{v}_{cm}.$$

While we can also add zeroth- and higher-order terms to the friction model, this is unnecessary when the object is free-flying (e.g. in space or in the air), as it is not in contact with a sliding surface.

We write the linear acceleration as

$$\mathbf{a}_{cm} = \frac{N d\mathbf{v}_{cm}}{dt} = \frac{B d\mathbf{v}_{cm}}{dt} + \boldsymbol{\omega} \times \mathbf{v}_{cm}.$$

Further, the rotational dynamics of B are given by $\mathbf{M}^{B/B_{cm}} = \mathbf{I}\boldsymbol{\alpha} + \boldsymbol{\omega} \times (\mathbf{I}\boldsymbol{\omega})$ where \mathbf{I} is the inertia matrix of B ,

$$\mathbf{I} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{bmatrix},$$

and $\mathbf{M}^{B/B_{cm}}$ is the resultant moment on B about the center of mass. $\mathbf{M}^{B/B_{cm}} = \mathbf{M}_f + \sum_{i=1}^n \mathbf{T}_i + \mathbf{r}_i \times \mathbf{F}_i$, where $\mathbf{M}_f = -\mu_r \boldsymbol{\omega}$ denotes the rotational friction on B .

Let $\mathbf{x} = \begin{bmatrix} v_x & v_y & v_z & \omega_x & \omega_y & \omega_z \end{bmatrix}^T$ denote the system state, and $\mathbf{u}_i = \begin{bmatrix} F_{xi} & F_{yi} & F_{zi} & T_{xi} & T_{yi} & T_{zi} \end{bmatrix}^T$ denote the input for agent i .

Using the above equations, we can write

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \sum_{i=1}^n \mathbf{B}_i(\mathbf{u}_i + \mathbf{L}_i \mathbf{f}(\mathbf{x})), \quad (2)$$

where

$$\mathbf{A} = \begin{bmatrix} -\frac{\mu_\ell}{m} \mathbb{I} & \mathbf{0} \\ \mathbf{0} & -\mu_r \mathbf{I}^{-1} \end{bmatrix}, \quad \mathbf{B}_i = \begin{bmatrix} \frac{1}{m} \mathbb{I} & \mathbf{0} \\ \mathbf{I}^{-1} \mathbf{R}_{i \times} & \mathbf{I}^{-1} \end{bmatrix},$$

$$\mathbf{L}_i = \frac{1}{n} \mathbf{B}_i^{-1} \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{P} \end{bmatrix}, \quad \mathbf{M} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{P} = \begin{bmatrix} 0 & I_{yz} & -I_{yz} & I_{xz} & -I_{xy} & I_{zz} - I_{yy} \\ -I_{xz} & 0 & I_{xz} & -I_{yz} & I_{xx} - I_{zz} & I_{xy} \\ I_{xy} & -I_{xy} & 0 & I_{yy} - I_{xx} & I_{yz} & -I_{xz} \end{bmatrix},$$

$$\text{and } \mathbf{f}(\mathbf{x}) = \begin{bmatrix} \omega_x v_y & \omega_x v_z & \omega_y v_x & \omega_y v_z & \omega_z v_x & \omega_z v_y & \cdots \\ \omega_x^2 & \omega_y^2 & \omega_z^2 & \omega_x \omega_y & \omega_x \omega_z & \omega_y \omega_z \end{bmatrix}^T,$$

where \mathbb{I} denotes the identity matrix, and $\mathbf{R}_{i \times}$ denotes the matrix form of the cross product of \mathbf{r}_i ,

$$\mathbf{R}_{i \times} = \begin{bmatrix} 0 & -r_{zi} & r_{yi} \\ r_{zi} & 0 & -r_{xi} \\ -r_{yi} & r_{xi} & 0 \end{bmatrix}.$$

Note the dynamics (1) and (2) are of the same general form.

IV. DECENTRALIZED ADAPTIVE CONTROL

In this section, we use techniques from Model Reference Adaptive Control (MRAC) to design decentralized adaptive controllers for a class of nonlinear systems. These controllers require no communication between agents, no prior knowledge of system parameters, and have proven stability and asymptotic tracking. We then use these results to develop adaptive controllers for the manipulation problems described in Section II.

A. Decentralized Adaptive Controller

Assume n agents are cooperating to control a nonlinear system, with system state $\mathbf{x} \in \mathbb{R}^m$. We assume the system is of the form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \sum_{i=1}^n \mathbf{B}_i(\mathbf{u}_i - \mathbf{L}_i \mathbf{f}_i(\mathbf{x})), \quad (3)$$

where $\mathbf{A} \in \mathbb{R}^{m \times m}$, $\mathbf{B}_i \in \mathbb{R}^{m \times k}$, $\mathbf{L}_i \in \mathbb{R}^{k \times q_i}$ are unknown constant matrices, and $\mathbf{f}_i(\mathbf{x}) \in \mathbb{R}^{q_i}$ are a set of known nonlinear basis functions which are continuous in \mathbf{x} . Note

this is precisely the form of the planar manipulation (1) and 3D manipulation (2) equations.

The objective of the decentralized controllers is to choose $\mathbf{u}_i \in \mathbb{R}^k$ such that the system state $\mathbf{x}(t)$ tracks the trajectory of a reference model $\mathbf{x}_m(t) \in \mathbb{R}^m$, with LTI dynamics given by

$$\dot{\mathbf{x}}_m = \mathbf{A}_m \mathbf{x}_m + \mathbf{B}_m \mathbf{r}(t), \quad (4)$$

where $\mathbf{A}_m \in \mathbb{R}^{m \times m}$ is stable, $\mathbf{B}_m \in \mathbb{R}^{m \times k}$, and $\mathbf{r}(t) \in \mathbb{R}^k$ is a bounded reference signal. The reference model is chosen by the designer, and represents the desired dynamics for the system.

Each agent i implements a controller of the form

$$\mathbf{u}_i = \mathbf{K}_{xi} \mathbf{x} + \mathbf{K}_{ri} \mathbf{r}(t) + \hat{\mathbf{L}}_i \mathbf{f}_i(\mathbf{x}), \quad (5)$$

where $\mathbf{K}_{xi} \in \mathbb{R}^{k \times m}$, $\mathbf{K}_{ri} \in \mathbb{R}^{k \times k}$, $\hat{\mathbf{L}}_i \in \mathbb{R}^{k \times q_i}$ are control gain matrices which are tuned online. Therefore, each agent seeks to choose control gains such that the closed-loop system tracks the reference model. This controller yields the closed-loop dynamics

$$\dot{\mathbf{x}} = (\mathbf{A} + \sum_{i=1}^n \mathbf{B}_i \mathbf{K}_{xi}) \mathbf{x} + (\sum_{i=1}^n \mathbf{B}_i \mathbf{K}_{ri}) \mathbf{r} + (\sum_{i=1}^n \mathbf{B}_i \tilde{\mathbf{L}}_i \mathbf{f}_i(\mathbf{x})), \quad (6)$$

where $\tilde{\mathbf{L}}_i = \hat{\mathbf{L}}_i - \mathbf{L}_i$.

Thus, if we choose \mathbf{K}_{xi}^* , \mathbf{K}_{ri}^* , \mathbf{L}_i^* according to

$$\begin{aligned} \mathbf{A} + \sum_{i=1}^n \mathbf{B}_i \mathbf{K}_{xi}^* &= \mathbf{A}_m, \\ \sum_{i=1}^n \mathbf{B}_i \mathbf{K}_{ri}^* &= \mathbf{B}_m, \\ \sum_{i=1}^n \mathbf{B}_i (\mathbf{L}_i^* - \mathbf{L}_i) \mathbf{f}_i(\mathbf{x}) &= 0, \end{aligned}$$

then the system dynamics match those of the reference model, and $\mathbf{x}(t) \rightarrow \mathbf{x}_m(t)$.

We assume the desired gains \mathbf{K}_{xi}^* , \mathbf{K}_{ri}^* , \mathbf{L}_i^* exist, although they are unknown *a priori*. Further, for the manipulation systems considered here, the desired gains always exist when the designer chooses $\mathbf{B}_m \succ 0$.¹

Define $\mathbf{e} = \mathbf{x} - \mathbf{x}_m$ as the tracking error. We use (4) and (6) to write the error dynamics

$$\begin{aligned} \dot{\mathbf{e}} &= \dot{\mathbf{x}} - \dot{\mathbf{x}}_m \\ &= \mathbf{A}_m \mathbf{e} + \sum_{i=1}^n \mathbf{B}_i (\tilde{\mathbf{K}}_{xi} \mathbf{x} + \tilde{\mathbf{K}}_{ri} \mathbf{r} + \tilde{\mathbf{L}}_i \mathbf{f}_i(\mathbf{x})), \end{aligned} \quad (7)$$

where $\tilde{\mathbf{K}}_{xi} = \mathbf{K}_{xi} - \mathbf{K}_{xi}^*$ is the feedback gain error and $\tilde{\mathbf{K}}_{ri} = \mathbf{K}_{ri} - \mathbf{K}_{ri}^*$ is the feedforward gain error.

Since the error dynamics depend on the unknown matrices \mathbf{B}_i , we make a further assumption on the form of \mathbf{K}_{ri}^* , namely:

Assumption 3. $\mathbf{K}_{ri}^* = \frac{1}{n} \mathbf{B}_i^{-1} \mathbf{B}_m$, for $\mathbf{K}_{ri}^* \succ 0$ or $\mathbf{K}_{ri}^* \prec 0$.

This assumption is non-restrictive, and implies the desired feedforward control effort is evenly divided between the agents. Since all \mathbf{B}_i are positive definite for the manipulation tasks considered in this paper (see Appendix), if $\mathbf{B}_m \succ 0$, then $\mathbf{K}_{ri}^* \succ 0$ by design.

¹We let $\mathbf{M} \succ 0$ denote that matrix \mathbf{M} is positive definite, and $\mathbf{M} \succ 0$ denote \mathbf{M} is negative definite.

We now propose the following adaptation laws for the control gain matrices:

$$\dot{\tilde{\mathbf{K}}}_{xi} = \dot{\mathbf{K}}_{xi} = -\Gamma_{xi} \mathbf{B}_m^T \mathbf{P} \mathbf{e} \mathbf{x}^T, \quad (8)$$

$$\dot{\tilde{\mathbf{K}}}_{ri} = \dot{\mathbf{K}}_{ri} = -\Gamma_{ri} \mathbf{B}_m^T \mathbf{P} \mathbf{e} \mathbf{r}^T, \quad (9)$$

$$\dot{\tilde{\mathbf{L}}}_i = \dot{\mathbf{L}}_i = -\Gamma_{li} \mathbf{B}_m^T \mathbf{P} \mathbf{e} \mathbf{f}_i(\mathbf{x})^T, \quad (10)$$

where $\mathbf{P} = \mathbf{P}^T \succ 0$ satisfies the Lyapunov equation

$$\mathbf{P} \mathbf{A}_m + \mathbf{A}_m^T \mathbf{P} = -\mathbf{Q}$$

for some $\mathbf{Q} = \mathbf{Q}^T \succ 0$, and $\Gamma_{xi}, \Gamma_{ri}, \Gamma_{li} \in \mathbb{R}^{k \times k}$ are positive definite gain matrices.

Theorem 1. *If each agent implements the controller in (5), and the adaptation laws (8)-(10), then the system is stable in the sense of Lyapunov, and achieves asymptotically perfect tracking.*

Proof. Consider the Lyapunov function candidate

$$V(\mathbf{e}, \tilde{\mathbf{K}}_{xi}, \tilde{\mathbf{K}}_{ri}, \tilde{\mathbf{L}}_i) = \mathbf{e}^T \mathbf{P} \mathbf{e} + \frac{1}{n} \sum \text{tr}(\tilde{\mathbf{K}}_{xi}^T \mathbf{K}_{ri}^{*-1} \Gamma_{xi}^{-1} \tilde{\mathbf{K}}_{xi} + \tilde{\mathbf{K}}_{ri}^T \mathbf{K}_{ri}^{*-1} \Gamma_{ri}^{-1} \tilde{\mathbf{K}}_{ri} + \tilde{\mathbf{L}}_i^T \mathbf{K}_{ri}^{*-1} \Gamma_{li}^{-1} \tilde{\mathbf{L}}_i).$$

Taking the time derivative of V yields

$$\begin{aligned} \dot{V} &= -\mathbf{e}^T \mathbf{Q} \mathbf{e} + \frac{2}{n} \sum \mathbf{e}^T \mathbf{P} \mathbf{B}_m \mathbf{K}_{ri}^{*-1} (\tilde{\mathbf{K}}_{xi} \mathbf{x} + \tilde{\mathbf{K}}_{ri} \mathbf{r} \\ &\quad + \tilde{\mathbf{L}}_i \mathbf{f}_i(\mathbf{x})) + \frac{2}{n} \sum \text{tr}(\tilde{\mathbf{K}}_{xi}^T \mathbf{K}_{ri}^{*-1} \Gamma_{xi}^{-1} \dot{\tilde{\mathbf{K}}}_{xi} \\ &\quad + \tilde{\mathbf{K}}_{ri}^T \mathbf{K}_{ri}^{*-1} \Gamma_{ri}^{-1} \dot{\tilde{\mathbf{K}}}_{ri} + \tilde{\mathbf{L}}_i^T \mathbf{K}_{ri}^{*-1} \Gamma_{li}^{-1} \dot{\tilde{\mathbf{L}}}_i). \end{aligned}$$

Using the adaptation laws (8)-(10), we get

$$\dot{V} = -\mathbf{e}^T \mathbf{Q} \mathbf{e}. \quad (11)$$

Thus, since $V > 0$ and $\dot{V} \leq 0$, the system is stable in the sense of Lyapunov, and $\mathbf{e}, \tilde{\mathbf{K}}_{xi}, \tilde{\mathbf{K}}_{ri}, \tilde{\mathbf{L}}_i$ are bounded because V is nonincreasing. By definition, $\mathbf{r}(t)$ is bounded, and since \mathbf{A}_m is a stable matrix, the reference model is BIBO stable, so \mathbf{x}_m is also bounded. Since \mathbf{e}, \mathbf{x}_m bounded, we also have \mathbf{x} bounded from the definition of \mathbf{e} , and \mathbf{u}_i bounded using (5) and the fact that \mathbf{f}_i are continuous in \mathbf{x} (and thus bounded for bounded \mathbf{x}). Therefore, all signals in the closed-loop are bounded. Further, from (7) we have $\dot{\mathbf{e}}$ bounded.

We can write the second derivative of V ,

$$\ddot{V} = -2\mathbf{e}^T \mathbf{Q} \dot{\mathbf{e}},$$

which is bounded since $\mathbf{e}, \dot{\mathbf{e}}$ bounded. Thus, we invoke Barbalat's Lemma [23] to conclude that $\dot{V} \rightarrow 0$ with t , and thus $\mathbf{e} \rightarrow 0$. \square

We note the controller (5) and adaptive laws (8)-(10) do not depend on n , the number of agents. Further, the controller is robust to the addition or removal of agents, since this is equivalent to merely restarting the controller with different n , and initial gains $\mathbf{K}_{xi}, \mathbf{K}_{ri}, \mathbf{L}_i$.

We also notice that the controller and adaptation laws require only information available to each agent (specifically $\mathbf{K}_{xi}, \mathbf{K}_{ri}, \mathbf{e}, \mathbf{r}, \mathbf{f}(\mathbf{x})$), without requiring information from any other agent. Thus, this is a decentralized adaptive controller that does not require explicit communication between robots.

Further, since (11) does not depend on $\tilde{\mathbf{K}}_{xi}$, $\tilde{\mathbf{K}}_{ri}$, $\tilde{\mathbf{L}}_i$, $\dot{V} \rightarrow 0$ does not imply the parameter errors go to zero asymptotically. An additional condition, persistent excitation (PE), must be satisfied for this to occur [19]. Thus, the control gains \mathbf{K}_{xi} , \mathbf{K}_{ri} , \mathbf{L}_i are not suitable for performing parameter estimation.

B. Decentralized Adaptive Manipulation

We now apply the results from Section IV-A to the manipulation problems formulated in Section II. We show our controller design solves these problems, and motivate Assumptions 1-2.

Corollary 1 (Controller for General Manipulation). *If a team of agents implement a decentralized controller of the form (5), and use adaptation laws of the form (8)-(10), they can solve Problem 1, namely for $\mathbf{x} = [\mathbf{v}_{cm}^T \ \boldsymbol{\omega}^T]^T$,*

$$\lim_{t \rightarrow \infty} \|\mathbf{x}(t) - \mathbf{x}_m\| = 0.$$

Proof. It is apparent that the dynamics of both systems in \mathbb{R}^2 (1) and \mathbb{R}^3 (2) are of the form in (3), with \mathbf{B}_i which satisfy Assumption 3 by definition (see Appendix). This result follows from Theorem 1. \square

We now motivate Assumptions 1-2 for the solution of Problem 1.

Assumption 1 is necessary because the controller (5) and adaptation law (8) require a measurement of \mathbf{x} , which requires a velocity measurement from the center of mass.

Further, the dynamics in (1), (2) implicitly assume the velocities are measured in a body-fixed frame. Thus, the agents must have access to this frame to achieve tracking.

Corollary 2 (Controller for Rotation Control). *If a team of agents have a shared body-fixed frame \hat{b}_{xyz} , and implement controllers of the form (5) with adaptation laws of the form (8)-(10), then they can solve Problem 2, namely for $\mathbf{x} = \boldsymbol{\omega}$,*

$$\lim_{t \rightarrow \infty} \|\mathbf{x}(t) - \mathbf{x}_m\| = 0.$$

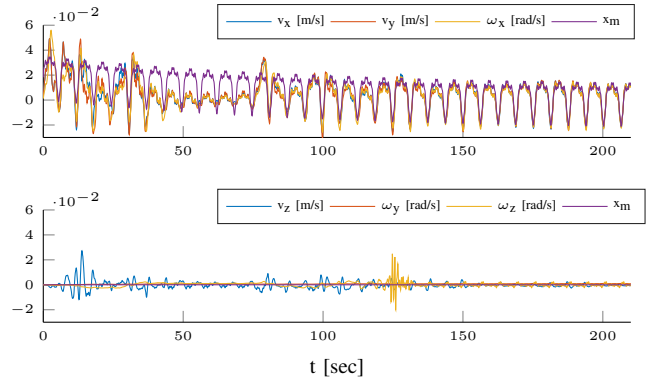
Proof. It is again apparent that the rotational dynamics of the systems are of the form (3). This result again follows from Theorem 1. \square

If the agents only apply torques \mathbf{T}_i to the object, none of the dynamics depend on, \mathbf{v}_{cm} . Further, since angular velocity is equal on all points of a rigid body, each agent can measure $\mathbf{x}(t)$ locally, which allows us to relax Assumption 1. However, solving Problem 2 still requires Assumption 2, since the agents require a common frame in which to express the reference signal \mathbf{r} .

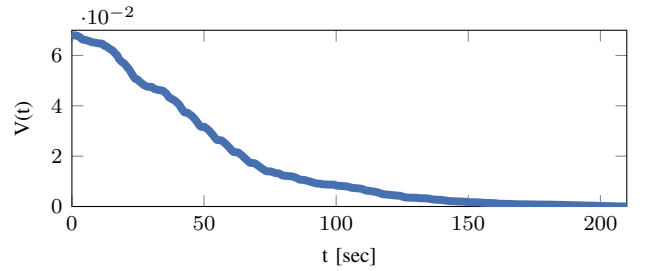
Corollary 3 (Controller for Rotation Stabilization). *If a team of agents implement a decentralized controller of the form (5) with adaptation laws of the form (8)-(10), they can solve Problem 3, namely*

$$\lim_{t \rightarrow \infty} \|\mathbf{x}(t)\| = 0.$$

Proof. Suppose $\mathbf{r}(t) = 0$. Since this reference signal is equal in all frames, the agents can implement the proposed adaptive controller without requiring Assumption 2. This result again follows from Theorem 1. \square



(a) Simulated Trajectory for General \mathbb{R}^3 Manipulation



(b) Lyapunov Function for General \mathbb{R}^3 Manipulation

Fig. 2: Simulation results for $n = 6$ agents manipulating a common payload in \mathbb{R}^3 . The agents aim to control the payload's angular and linear velocities. (a): Simulated state trajectory, and reference trajectory. (b): Simulated Lyapunov function V for the manipulation task. The plotted function has been shifted by a constant value.

Parameter	Value
m	2.6e4 (kg)
I_{xx}	4.47e6 (kg m ²)
I_{yy}	4.47e6 (kg m ²)
I_{zz}	1.40e5 (kg m ²)
\mathbf{r}_1	(0, 0, 20.4) (m)
\mathbf{r}_2	(6.9, 0, 0) (m)
\mathbf{r}_3	(-6.9, 0, 0) (m)
\mathbf{r}_4	(0, 6.9, 0) (m)
\mathbf{r}_5	(0, -6.9, 0) (m)
\mathbf{r}_6	(0, 0, -20.4) (m)

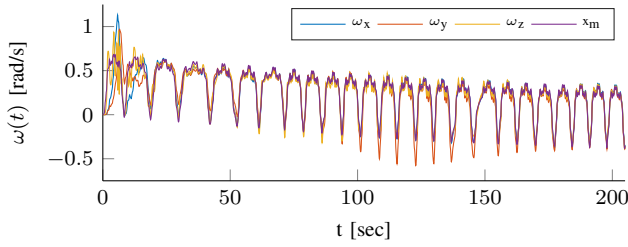
TABLE I: Simulation Parameters

V. SIMULATION RESULTS

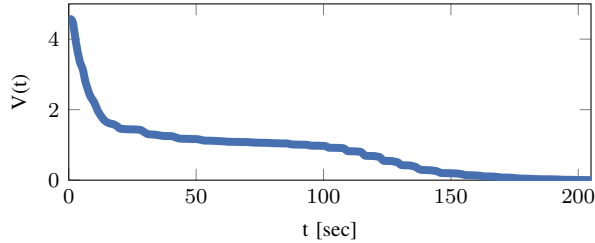
To validate our algorithm, we simulated its performance for Problems 1-3. The simulation consisted of a team of $n = 6$ satellites rigidly attached to a large, free-floating payload (i.e. a rocket body). The mass properties of the payload were based on a Delta IV first stage rocket, and the model parameters are given in Table I.

While the adaptation laws (8)-(10) are continuous derivatives, we used a forward-difference method to approximate the derivative for a discrete-time implementation. Additionally, we modified the adaptation laws by adding σ -modification and a deadzone to improve their robustness and transient performance. These techniques are well studied, and outlined in [24].

Figure 2(a) shows the simulated trajectory for a group of agents performing general manipulation on the body B . A sinusoidal reference signal of multiple frequencies was



(a) Simulated Trajectory for \mathbf{R}^3 Rotation Control



(b) Lyapunov Function for \mathbf{R}^3 Rotation Control

Fig. 3: Simulation results for $n = 6$ agents controlling the rotation of a payload in \mathbf{R}^3 . Only local measurements are used for control. (a): Simulated state trajectory, and reference trajectory. (b): Simulated Lyapunov function V for the manipulation task. The plotted function has been shifted by a constant value.

commanded in v_x , v_y , and ω_x , while zero velocities were commanded in the other states. The agents were able to measure the velocities at the center of mass, and had a common body-fixed frame. It is apparent that the system achieves asymptotic tracking, as shown in Corollary 1.

Further, Figure 2(b) plots the Lyapunov function $V(t)$ for the simulation. For ease of computation, the function $V(t)$ was evaluated by integrating \dot{V} , given by (11). Since $V \rightarrow 0$ with t , the plot is offset by a constant amount. From the plot, we can see $\dot{V} \rightarrow 0$.

Figure 3(a) shows the simulated trajectories for a group of agents controlling the rotation of body B . The agents only applied torques \mathbf{T}_i to the body, and used local measurements for control. The figure indicates that asymptotic tracking is achieved. Further, from Figure 3(b), we again see that $\dot{V} \rightarrow 0$.

As discussed in Section IV-B, the control strategy is robust to the addition/removal of agents during manipulation. Figure 4 shows a simulation in which the agents attempt to control the rotation of body B , where 4 of the $n = 6$ agents are turned off at $t_{drop} = 45$ s. The controllers are able to respond to this change, and still achieve asymptotic tracking.

Further, Figure 5 plots the simulated trajectories of a group of agents attempting to stabilize body B to zero angular velocity. The body was given an initial angular velocity of $[3, 3, 3]$ (rad/s) about its principal axes. The agents did not have access to central measurements or a common reference frame. Despite these constraints, the agents were still able to stabilize the body's rotation.

VI. EXPERIMENTAL RESULTS

We also evaluated the control strategy experimentally, using a team of ground-based mobile robots. A schematic of the experimental setup is shown in Figure 6. The robots have omnidirectional wheels, and use an onboard PID controller

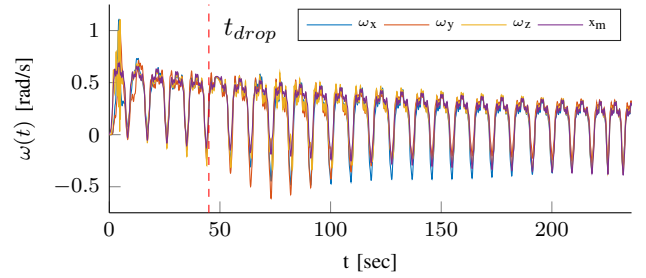


Fig. 4: Simulated trajectory for $n = 6$ agents controlling the rotation of a common payload. At $t_{drop} = 45$ s, 4 agents are deactivated.

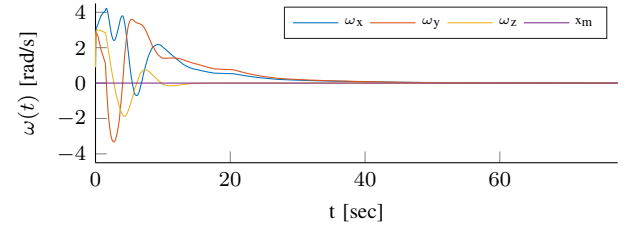


Fig. 5: Simulated trajectory for $n = 6$ agents attempting to stabilize the rotation of a body in \mathbf{R}^3 . Only local measurements are used for control, and agents do not have a common reference frame.

to provide the desired force \mathbf{F}_i and torque \mathbf{T}_i by generating motor speeds. The robots received velocity measurements via an OptiTrack motion capture system. All controller code was executed onboard each robot. All communication was performed using ROS over a wireless network. The payload had a mass of $m = 16.5$ kg, and inertia $J = 2.46$ kg m^2 .

Figure 7(a) shows the results of the controllers when commanded a square wave in v_x , and zero velocity in v_y or ω . The test consists of 50 trials of 60 seconds each. Each robot was initialized with randomly scaled control gains, and had no unique prior knowledge of the object parameters or its position on the object. While the robots initially do not respond to the velocity command, their controllers adapt online to allow the payload to begin moving, and eventually to track the reference signal.

Figure 7(b) shows the average value of the Lyapunov function $V(t)$ for the trials, as well as the maximum and minimum value at each timestep. The variance in the function values is likely due to unmodelled effects, including network latency, discretization error, and static friction. It is apparent that \dot{V} is decreasing in magnitude, but does not fully reach 0 by the end of the trial, indicating the parameters can evolve further to achieve better tracking.

Figure 7(c) shows the results of the controllers when tracking a reference signal generated by a human user. The reference signal was generated in real time using a joystick operated by the human user, and broadcast over the wireless network.

Again, we can see the controllers initially do not respond to velocity commands, but begin to track the commands as their controllers adapt. Good tracking is achieved for arbitrary signals in v_x , v_y , while moderate tracking is achieved in ω . The difficulty in rotational control is likely due to static rotational friction, as well as control gains which were initialized to values far from their true values.

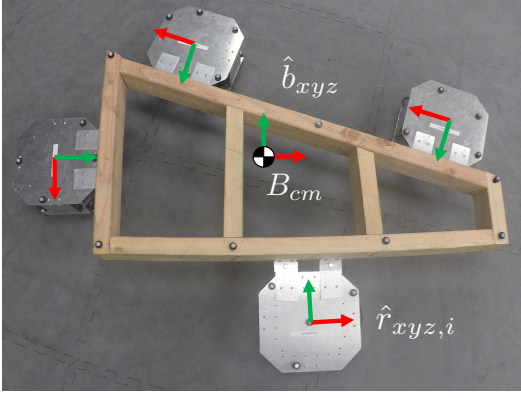


Fig. 6: Four ground robots manipulate a common payload. The agents achieve asymptotic tracking in linear and angular velocity without prior knowledge or explicit communication with other agents. All computation is performed onboard.

VII. CONCLUSION AND FUTURE WORK

This work proposes a decentralized adaptive controller which allows teams of agents to perform collaborative manipulation without communication or prior knowledge of object parameters. The proposed controller has proven stability and tracking for linear and angular velocities in \mathbb{R}^2 and \mathbb{R}^3 , if the agents receive measurements from the center of mass, tracking in angular velocity if agents have a common body-fixed frame, and rotation stabilization using only local measurements. The strategy was verified in both simulation and experiments.

A number of improvements can be made to the proposed control strategy. Assumption 1, which requires center of mass measurements for general manipulation, can be relaxed by formulating the object dynamics about a general point on the rigid body, which would allow a sensor to be placed anywhere on the body, reducing the need for object calibration.

Further, while the proposed controllers have proven tracking in $\mathbf{x}(t)$, the parameter errors do not necessarily go to zero. An additional condition, persistent excitation (PE), must hold for the parameters to converge to their true values [19]. Thus, for general $\mathbf{r}(t)$, the controller gains are not suitable for parameter estimation. Future work includes writing the conditions for PE and investigating when they are satisfied. We also hope to verify our manipulation algorithms in \mathbb{R}^3 experimentally using quadrotors or robotic arms.

VIII. ACKNOWLEDGEMENTS

The authors would like to thank Zijian Wang for his invaluable help and advice while conducting experiments with the Oujabots.

APPENDIX

Here we prove the positive definiteness of the matrices \mathbf{B}_i as defined in Section III.

Lemma 1 (Definiteness of Block Triangular Matrices). *For a matrix $\mathbf{M} \in \mathbb{R}^{(f+g) \times (f+g)}$ of the form*

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C} & \mathbf{D} \end{bmatrix},$$

where $\mathbf{A} \in \mathbb{R}^{f \times f}$, $\mathbf{C} \in \mathbb{R}^{g \times f}$, $\mathbf{D} \in \mathbb{R}^{g \times g}$, if $\mathbf{A}, \mathbf{D} \succ 0$ then $\mathbf{M} \succ 0$.

Proof. Using the results in [25], we write

$$\det(\mathbf{M} - \lambda \mathbf{I}) = \det(\mathbf{A} - \lambda \mathbf{I}) \det(\mathbf{D} - \lambda \mathbf{I})$$

Thus, the eigenvalues of \mathbf{M} are equal to the union of those of \mathbf{A} and \mathbf{D} , which have positive real parts. Thus, $\mathbf{M} \succ 0$, as needed. \square

Theorem 2. \mathbf{B}_i , as defined in (1) is positive definite.

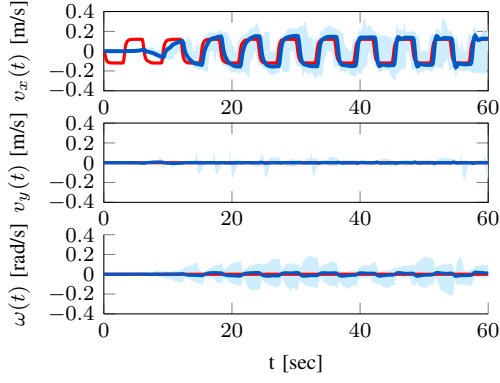
Proof. We can partition \mathbf{B}_i with $\mathbf{A} = \frac{1}{m} \mathbf{I}$ and $\mathbf{D} = \frac{1}{J} \mathbf{I}$. By inspection, $\mathbf{A}, \mathbf{D} \succ 0$. Thus, using Lemma 1, $\mathbf{B}_i \succ 0$, as needed. \square

Theorem 3. \mathbf{B}_i , as defined in (2) is positive definite.

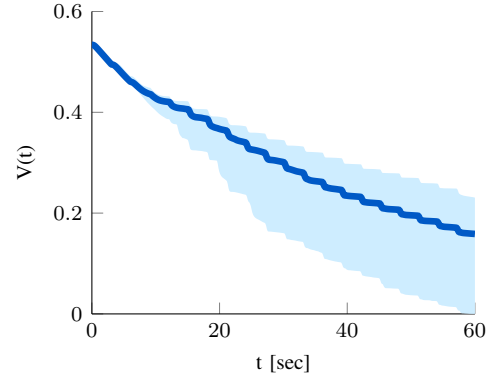
Proof. We again partition \mathbf{B}_i , with $\mathbf{A} = \frac{1}{m} \mathbf{I}$ and $\mathbf{D} = \mathbf{I}^{-1}$. By inspection, $\mathbf{A} \succ 0$. For rigid bodies of finite size, $\mathbf{I} \succ 0$, which implies $\mathbf{I}^{-1} \succ 0$, and thus $\mathbf{D} \succ 0$. Thus, by Lemma 1, $\mathbf{B}_i \succ 0$, as required. \square

REFERENCES

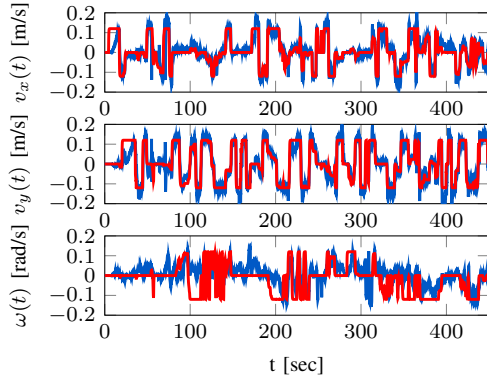
- [1] Z. Wang, G. Yang, X. Su, and M. Schwager, "Oujabots: Omnidirectional robots for cooperative object transport with rotation control using no communication," in *Proc. Int. Conf. Distrib. Auton. Robot. Syst.*, 2016.
- [2] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3. IEEE, pp. 2149–2154.
- [3] 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*, vol. 1. IEEE Comput. Soc. Press, pp. 235–242.
- [4] 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*, vol. 13, no. 11, pp. 755–764, Nov 1996.
- [5] 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)*, vol. 2. IEEE, pp. 1217–1222.
- [6] 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*. IEEE, May 2008, pp. 1471–1476.
- [7] Z. Wang and M. Schwager, "Force-Amplifying N-robot Transport System (Force-ANTS) for cooperative planar manipulation without communication," *The International Journal of Robotics Research*, vol. 35, no. 13, pp. 1564–1586, Nov 2016.
- [8] S. G. Faal, S. T. Kalat, and C. D. Onal, "Towards collective manipulation without inter-agent communication," in *Proceedings of the 31st Annual ACM Symposium on Applied Computing - SAC '16*. New York, New York, USA: ACM Press, 2016, pp. 275–280.
- [9] R. A. Knepper, T. Layton, J. Romanishin, and D. Rus, "IkeaBot: An autonomous multi-robot coordinated furniture assembly system," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, May 2013, pp. 855–862.
- [10] J. Alonso-Mora, R. Knepper, R. Siegwart, and D. Rus, "Local motion planning for collaborative multi-robot manipulation of deformable objects," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2015, pp. 5495–5502.
- [11] K. Sreenath and V. 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, Jun 2013.



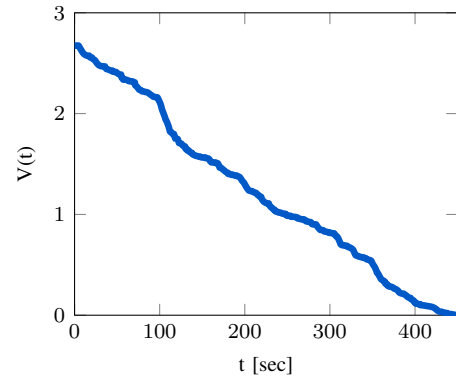
(a) Trajectory, Square Wave in v_x , 50 Trials



(b) Lyapunov Function, Square Wave in v_x , 50 Trials



(c) Trajectory, User-Generated Reference



(d) Lyapunov Function, User-Generated Reference

Fig. 7: Experimental results for manipulation in \mathbb{R}^2 . (a): An average trajectory (dark blue) over 50 trials for a reference trajectory (red) consisting of a square wave in v_x , and zeros in v_y, ω . The range of observed trajectories is filled in light blue. (b): Lyapunov function (dark blue) for the experiment in (a), averaged over 50 trials, with range of values shown in light blue. (c): State trajectory for reference generated by human user with a joystick. (d): Lyapunov function for experiment in (c).

- [12] D. Mellinger, M. Shomin, N. Michael, and V. Kumar, "Cooperative Grasping and Transport Using Multiple Quadrotors," in *Distributed Autonomous Robotic Systems: The 10th International Symposium*, A. Martinoli, F. Mondada, N. Correll, G. Mermoud, M. Egerstedt, M. A. Hsieh, L. E. Parker, and K. Støy, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 545–558.
- [13] H. Lee, H. Kim, and H. J. Kim, "Planning and Control for Collision-Free Cooperative Aerial Transportation," *IEEE Transactions on Automation Science and Engineering*, pp. 1–13, 2016.
- [14] 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)*. IEEE, May 2015, pp. 5517–5522.
- [15] 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)*. IEEE, May 2016, pp. 441–446.
- [16] T. Lee, "Robust adaptive attitude tracking on $SO(3)$ with an application to a quadrotor uav," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 5, pp. 1924–1930, 2013.
- [17] Y.-H. Liu and S. Arimoto, "Decentralized Adaptive and Nonadaptive Position/Force Controllers for Redundant Manipulators in Cooperations," *The International Journal of Robotics Research*, vol. 17, no. 3, pp. 232–247, Mar 1998.
- [18] Z. Li, S. S. Ge, and Z. Wang, "Robust adaptive control of coordinated multiple mobile manipulators," *Mechatronics*, vol. 18, no. 5-6, pp. 239–250, Jun 2008.
- [19] P. A. Ioannou and J. Sun, *Robust adaptive control*. PTR Prentice-Hall Upper Saddle River, NJ, 1996, vol. 1.
- [20] P. Ioannou, "Decentralized adaptive control of interconnected systems," *IEEE Transactions on Automatic Control*, vol. 31, no. 4, pp. 291–298, Apr 1986.
- [21] K. Narendra and N. Oleng, "Decentralized adaptive control," in *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*. IEEE, 2002, pp. 3407–3412 vol.5.
- [22] J.-C. Liou, N. Johnson, and N. Hill, "Controlling the growth of future leo debris populations with active debris removal," *Acta Astronautica*, vol. 66, no. 5, pp. 648 – 653, 2010.
- [23] J.-J. E. Slotine, W. Li *et al.*, *Applied nonlinear control*. Prentice hall Englewood Cliffs, NJ, 1991, vol. 199, no. 1.
- [24] E. Lavretsky and K. A. Wise, "Robust Adaptive Control." Springer, London, 2013, pp. 317–353.
- [25] J. R. Silvester, "Determinants of block matrices," *The Mathematical Gazette*, vol. 84, no. 501, pp. 460–467, 2000.