# Multi-Robot Manipulation without Communication

Zijian Wang and Mac Schwager

**Abstract** This paper presents a novel multi-robot manipulation algorithm which allows a large number of small robots to move a comparatively large object along a desired trajectory to a goal location. The algorithm does not require an explicit communication network among the robots. Instead, the robots coordinate their actions through sensing the motion of the object itself. It is proven that this implicit information is sufficient to synchronize the forces applied by the robots. A leader robot then steers the forces of the synchronized group to manipulate the object through the desired trajectory to the goal. The paper presents algorithms that are proven to control both translational and rotational motion of the object. Simulations demonstrate the approach for two scenarios with 20 robots transporting a rectangular plank and 1000 robots transporting a piano.

## 1 Introduction

In this paper we present a scalable, decentralized control strategy by which a large number of robots can manipulate a comparatively large object through a desired trajectory to a goal configuration. The key to the approach is to use the object itself as a medium for transferring information throughout the group of robots. No communication network is required in this strategy. Instead, the robots sense the local motion of the object, and use this information to correct their own force through a feedback law. We prove that this feedback law will cause all robots' forces to align to the same direction exponentially fast. Furthermore, the rate of this exponential convergence increases linearly with the number of robots, so that performance becomes *faster* as the number of robots *increases*, leading to a scalable strategy.

---

Zijian Wang, Mac Schwager
Department of Mechanical Engineering, Boston University, Boston, MA, USA
e-mail: {zjwang, schwager}@bu.edu

The forces of the robots synchronize to a leader, which can either be a robot or a human operator. The leader then guides the object through a desired trajectory to the goal configuration, using the synchronized follower robots to multiply its effective force. We provide feedback control laws for the leader to steer the whole system, both in translation and in rotation, under a mild symmetry assumption on the follower robots. We require the leader to know the object's location relative to the desired trajectory, however the follower robots do not need to know their location, the locations of other robots, the object, or the desired trajectory. We demonstrate the approach in simulations with two scenarios, one involving 20 robots manipulating a plank of wood, and another with 1000 robots manipulating a piano.

Our algorithm is useful in situations where a large number of small, inexpensive robots are required to coordinate to manipulate large objects. For example, in an automated construction site our system could be used to transport massive building materials to a desired location. In a manufacturing facility, this system could be used to transport large products (e.g. aircraft, trains, or industrial equipment) in various stages of assembly to different areas on the manufacturing floor. In our system the leader can be a human operator, enabling one person to move objects that would otherwise require a forklift or a crane. Similarly, in a disaster relief scenario, such a system of robots could be used to autonomously clear debris from a collapsed building to free survivors, or to clean up structurally unstable disaster sites.

The most attractive aspect of our approach is that no explicit communication is needed among robots. Coordinated control algorithms that rely on a wireless network must deal with dropped packets, packet collisions, delays, and fundamental scaling capacity limitations [18]. The presence of a network also requires more sophisticated robots with more sophisticated hardware. In contrast, we take a minimalist approach to achieve scalability. By sensing the motion of the object itself, our follower robots can determine the summed forces of all the other robots acting on the object. This is all the information needed to reach a consensus on the robots' forces. Hence our robots do not require networking hardware, localization information, nor a global reference frame.

## 1.1 Related Work

Manipulation is a fundamental problem in robotics with an enormous literature, and many algorithms for multi-robot manipulation have been proposed. An early approach to multi-robot manipulation can be found in [15, 16], where various pushing strategies are designed given different availability of sensing and communication. Another approach, known as caging was studied in [1], where a group of robots surrounds an object and then moves together, making sure that the object always remains inside the formation. In this vein, some work has focused on a rigorous geometrical analysis of how the object can be completely caged [2, 3, 4]. Although mathematically elegant, these methods typically have considerable computational requirements [5]. On the other hand, some authors have studied formation-based caging, which assumes that there are enough robots so that the object cannot escape the formation. The object can then be transported by moving the formation as a rigid body using, for example, potential fields [6] or vector fields [7]. Other approaches have used novel modes of actuation, for example tow cables [17, 19]. The approach we describe here is different from all of these in that we require no explicit communication between robots. In this respect, our algorithm is most similar to ensemble control techniques from [8, 9], which also do not require robot-to-robot communication. Our technique is different from ensemble control in that each robot individually steers itself according to its own control actions, as opposed to having one common control signal for the whole group.

The analysis of our control strategy takes some inspiration from the study of multi-agent consensus [10, 11] and the study of leader-follower networks [20, 21]. In consensus problems, agents locally exchange information about their neighbors' states through a communication network, in order to reach a consensus on a quantity of interest. Similarly, in leader-follower networks, local communication protocols are used for all agents to converge to the state of a leader. As opposed to these works, we do not have an explicit communication network, however we use analytical tools from this work to show that the robots in our system will converge to the force of the leader robot.

Our work is also inspired by collective ant transport strategies studied in Behavioral Biology and Entomology. Ants, like our robots, have no wireless network, yet they are able to effectively coordinate their actions to manipulate large objects. It was hypothesized in [13] that ants detect small-scale vibration or deformation of the object in order to coordinate their forces. Our algorithm suggests an even simpler hypothesis: ants might synchronize their actions using only the rigid body motion of the object they are trying to transport. In [12] the authors measured the forces exerted by a group of ants and found that ants aligned their forces better and better as the manipulation task went on, which agrees with the synchronization approach we propose here. In addition to translation, [14] determined that only a small number of ants in the group are crucial for the rotation of the object, which is similar to the role of the leader robot in our approach.

The rest of this paper is organized as follows. We model the physics of the object and robots and formally state the problem in Sec. 2. In Sec. 3 we present our multi-robot control strategy for both the followers and the leader and analyze its convergence properties. Finally, in Sec. 4 we show numerical simulations with 20 robots and 1000 robots, and we give our conclusions in Sec. 5.

## 2 Modeling and Problem Formulation

We consider a planar region $Q \subset \mathbb{R}^2$, where there is a target object with mass $M$ and moment of inertia $J$, as shown in Figure 1. The object has three degrees of freedom, that is, the position of the center of mass $x_c \in \mathbb{R}^2$ and the orientation $\theta \in SO(2)$. There is friction force in $Q$, and we model it as the sum of static friction and viscous friction, whose coefficients are represented as $\mu_s$ and $\mu_v$, and the acceleration of gravity is $g$.

We have a group of $N$ identical robots $R_i,\ i \in \{1, 2, \cdots, N\}$, trying to transport the object from its initial position to some destination in $Q$. Each robot is capable of: (i) gripping the object at $x_i$, which is on the edge of the object, and applying a force $F_i \in \mathbb{R}^2$ in any direction and with any magnitude below its maximum force limit; (ii) measuring the velocity and acceleration, denoted by $v = \dot{x}_c$ and $\dot{v}$, of the manipulated object in the robot's local reference frame;[1] (iii) following the movement of the object such that the desired force can be maintained. We also have a leader robot, indexed as $R_1$, that is more powerful than the rest of follower robots in that: (i) in addition to applying a force, the leader can also apply a torque $T_1 \in \mathbb{R}$ to the object; (ii) the leader can measure the angle and angular velocity of the object; (iii) the leader knows the desired trajectory, and it can also measure the relative position between the trajectory and the object.

Under the forces from the robots and the environment, the object will move with translational velocity $v$ and angular velocity $\omega$. In the next section, we will derive the translational and rotational dynamics of the object, and then state our problem formally.

## *2.1 Translational Dynamics*

For translation, the forces are either friction or from robots. So the translational dynamics can be written according to Newton's second law

$$M\dot{v} = \sum_{i=1}^{N} F_i - \mu_s M g \frac{v}{\|v\|} - \mu_v v. \tag{1}$$

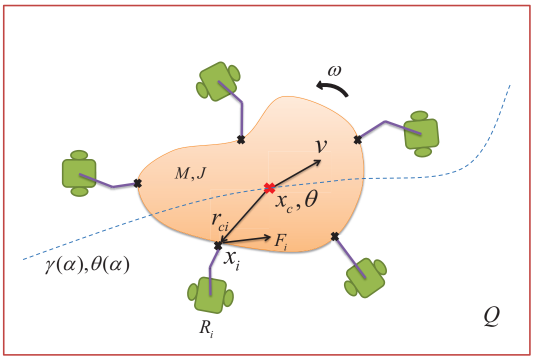


**Fig. 1** Configuration of our multi-robot manipulation task. The figure shows an example where five robots (in green) are manipulating an object (in red).

[1] This requirement can be relaxed so that robots only know the velocity and acceleration of their attachment point, $\dot{x}_i$ and $\ddot{x}_i$, in their local reference frame, however this considerably complicates the dynamics. We treat the simpler case here for clarity.

## 2.2 Rotational Dynamics

Since the object has geometric extension around the center of mass rather than a point mass, the force applied to the object will generate a torque if it does not point to or from the center of mass. In order to characterize the rotational dynamics, we need to study two types of torques, associated with friction and robots' forces respectively.

**Frictional Torque.** Here we derive a model of the torque due to viscous friction on the object. We neglect any torque due to static friction, as such effects are difficult to model, and are expected to be small in comparison to viscous and inertial forces. Consider the velocity of an arbitrary point $x$ on the object, which is translating while rotating as shown in Figure 2,

$$v_a = v + \omega \times r,$$

where $v_a$ is the absolute velocity, $v$ is the translational velocity at $x_c$, $r$ is the vector pointing from $x_c$ to our selected point. Note that $v_a$ is different for different points on the object. Then the viscous friction at any selected point can be written as

$$F_v = -\mu_v v_a = -\mu_v v - \mu_v(\omega \times r). \tag{2}$$

The total frictional torque can then be calculated by taking the integral of (2) over the object's bottom surface. Our conclusion is that the torque generated by the viscous friction is proportional to the angular velocity, as shown below.

**Proposition 1.** ***(Frictional Torque)*** *Given an object with arbitrary shape in Q, denote the torque caused by the viscous friction by $T_f$. Then*

$$T_f = -\frac{\mu_v}{M} J\omega. \tag{3}$$

*Proof.* In (2), the viscous friction $F_v$ has two parts, $-\mu_v v$ and $-\mu_v(\omega \times r)$. Denote the torques associated with them by $T_{f1}$ and $T_{f2}$ respectively. Denote density of the object by $\rho$, so we know the center of mass $x_c$ can be represented as

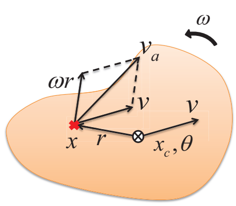


Fig. 2: The synthesis of the absolute velocity of an arbitrary point on the object.

$$x_c = \frac{\int_S \rho x dx}{\int_S \rho dx} = \frac{\int_S \rho x dx}{M}.$$

Then we have

$$\begin{aligned}
T_{f1} &= -\int_S \frac{\rho \mu_v}{M}(r \times v_e)dr = -\int_S \frac{\rho \mu_v}{M}\left[(x - x_c) \times v_e\right]dx \\
&= -\frac{\mu_v}{M}\int_S \rho x \times v_e dx + \frac{\mu_v}{M} x_c \times v_e \int_S \rho dx = -\frac{\mu_v}{M}\left(\int_S \rho x dx\right) \times v_e + \mu_v x_c \times v_e \\
&= -\frac{\mu_v}{M} M x_c \times v_e + \mu_v x_c \times v_e = 0.
\end{aligned}$$

As for $T_{f2}$, note that in the object's local reference frame, $\omega$ is always perpendicular to $r$, so we have

$$T_{f2} = -\int_S \frac{\rho \mu_v}{M}(\omega \times r) r dr = -\int_S \frac{\rho \mu_v}{M} \omega r^2 dr = -\left(\frac{\mu_v}{M}\int_S \rho r^2 dr\right)\omega = -\frac{\mu_v}{M} J \omega.$$

Finally we have $T_f = T_{f1} + T_{f2} = T_{f2} = -\frac{\mu_v}{M} J \omega$. □

**Robots' Torques.** Torques from robots also have two parts. The first part comes from the forces applied by every robot on the edges of the object. The second part comes from the leader robot's direct torque input. The robots cannot compute the first part because we assume that they do not know their position on the object, $r_{ci}$. However, with many robots equally spaced around the perimeter, these torques would cancel out exponentially fast. This is formalized in the following symmetry assumption.

**Assumption 1** ***(Centrosymmetric)****. The distribution of the positions of robots' forces is centrosymmetric around $x_c$. In other words, $\sum_{i=1}^{N} r_{ci} = 0$.*

Under Assumption 1, when consensus is reached, all $F_i$ will be the same and thus the resulting torque is zero, which is shown in Figure 3. In contrast, in Figure 4 where the forces are not centrosymmetric, the torque is not necessarily zero. In realistic situations, if the number of robots is large with respect to the size of the object, then it is more likely that Assumption 1 will be satisfied or nearly satisfied. However, Assumption 1 does not guarantee that the torque caused by all $F_i$ will always be zero. For example, when the leader's force changes dramatically in a turning process, it will take some time for followers to track the leader and reach the consensus although this process is exponentially fast. We view it as the modeling error, which can be dealt with by the robustness of our controller.

Therefore, the only torque we consider from robots is the leader's direct torque input $T_1$. Hence, the overall rotational dynamics can be written as

$$J\dot{\omega} = T_1 - T_f = T_1 - \frac{\mu_v}{M} J \omega. \tag{4}$$

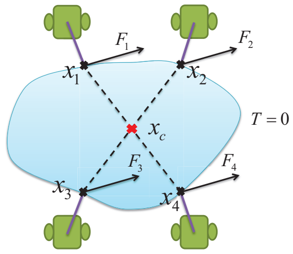


Fig. 3: Centrosymmetric forces.

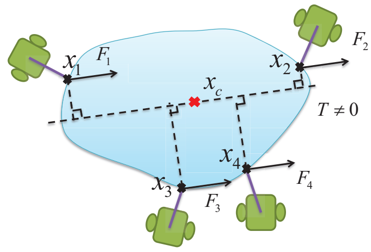


Fig. 4: Non-centrosymmetric forces.

### *2.3 Problem Formulation*

We assume that the object is too heavy for any individual robot to move due to the static friction force. However, if all the forces from every robot are aligned, the sum of these forces can overcome the static friction, i.e.,

$$max\{\|F_i\|\} < \mu_s Mg \tag{5}$$

$$\frac{\mu_s Mg}{max\{\|F_i\|\}} < N. \tag{6}$$

The goal is to coordinate the forces from all robots to transport the object along a desired trajectory. We define the desired trajectory as

$$\begin{aligned} \gamma(\alpha) &: [0,1] \mapsto Q \\ \theta &= g_\theta(\gamma), \end{aligned} \tag{7}$$

where $\alpha$ is the index of the point on the trajectory while $g_\theta$ is the function that specifies the desired angles for different points on the trajectory. For example, $\gamma(\alpha = 0)$ is the start of the trajectory, and $\gamma(\alpha = 1)$ is the end of the trajectory. Note that the desired trajectory is given by other path planners, such as [22].

Robots only know parameters $M, \mu_s, \mu_v, g, N$, but they do not know the global position of the object, their own, other robots, nor where they attach to the object, i.e., $x_i$. There is no explicit communication between any two robots and the desired trajectory is only known to the leader robot.

## 3 Control Strategy and Analysis

In this section, we will present how to coordinate the forces from all robots using consensus but without communication. Using consensus, we can transform the original $2N$-dimensional force into a reduced state representation. Furthermore, we treat the leader robot's force and torque as the inputs of the entire system while the linear and angular velocity of the object are the outputs. The overall dynamics is studied and enables us to design the controller for trajectory following based on it.

### *3.1 Force Coordination With Consensus*

Consensus in our case means that all robots will eventually apply the same forces to the object. In conventional consensus methods, explicit communication is required among agents to exchange state information. In contrast, we avoid explicit communication by using the local measurement of the object's movement.

At the beginning of the task, the motion of the object can be initiated randomly.[2] Once the object starts to move, all the robots use the following force updating law

$$\begin{aligned}\dot{F}_i(t) &= \sum_{j=1, j\neq i}^{N} \big(F_j(t) - F_i(t)\big) \\ &= \sum_{j=1}^{N} F_j(t) - NF_i(t) = M\dot{v} + \mu_s Mg\frac{v}{\|v\|} + \mu_v v - NF_i(t).\end{aligned} \tag{8}$$

Eq. (8) is computable by every robot since all the terms are locally known without communication. Note that in practice, the readings of $v$ and $\dot{v}$ are in each robot's local reference frame. However, this does not affect our algorithm, therefore we do not distinguish between $v$ and $\dot{v}$ in the local and global reference frames.

The first row in (8) is the commonly used consensus protocol. If we stack all the forces into one vector $F(t) = (F_1(t)\ F_2(t)\ \cdots\ F_N(t))^T$, then (8) can be also put into the matrix form

$$\dot{F}(t) = -LF(t) \tag{9}$$

$$L = \begin{pmatrix} N-1 & -1 & \cdots & -1 \\ -1 & N-1 & \cdots & -1 \\ \vdots & \vdots & \ddots & -1 \\ -1 & -1 & \cdots & N-1 \end{pmatrix}, \tag{10}$$

where we can find out that $L$ is the graph Laplacian of a completely connected graph. Since $-L$ is negative semi-definite and (9) is a stable linear system, we have that $F(t)$ will converge to the null space of $L$, which is spanned by $\mathbf{1}$. More specifically, $F(t)$ will converge to $Ave\big(F(0)\big)\mathbf{1} = \big(\sum_{i=1}^{N} F_i(0)/N\big)\mathbf{1}$, as proven in [10]. This is how we produce the consensus without communication.

The force updating law (8) ensures the force consensus ending up with the average value of the initial forces. However, we also want to steer the consensus in order to get any desired manipulation force for trajectory following. In order to do this, we let the leader robot $R_1$ make its own decision about what force to apply. If the

[2] For example, all the robots can repeatedly apply forces in random directions. Eventually enough of the forces will algin by chance to overcome static friction, and the object will begin to move.

leader does not change its value, then all the followers will converge to the leader's value, i.e.,

$$\lim_{t\to\infty} F(t) = F_1(0)\mathbf{1}, \tag{11}$$

where $F_1(0) \in \mathbb{R}^2$ stands for the leader's force, as proven in [11]. More generally, if the leader keeps changing its value, the followers will still follow the leader, and we will discuss this in detail in the next section.

## *3.2 Reduced State Representation*

Here we present a reduced state representation in the following theorem. Having the reduced state representation, the changing leader can know how the followers will follow it and what the group force will be, which we will use to control the object's movement.

**Theorem 1.** ***(Reduced State Representation)*** *Given a multi-robot system containing N robots, let robot $R_1$ be the only leader in the group, and the rest are followers which update their forces using (8). Then the reduced state representation can be written as:*

$$\begin{aligned} \dot{\eta}(t) &= -\eta(t) + F_1(t) \\ F_s(t) &= (N-1)\eta(t) + F_1(t) \end{aligned} \tag{12}$$

*where $F_s(t) = \sum_{i=1}^{N} F_i(t)$ is the group force, and $\eta(t) = (\sum_{i=2}^{N} F_i(t))/(N-1)$ denotes the average force of all followers.*

*Proof.* By adding up (8) when $i$ goes from 2 to $N$ we have

$$\begin{aligned} \sum_{i=2}^{N} \dot{F}_i(t) &= (N-1)\sum_{j=1}^{N} F_j(t) - N\sum_{i=2}^{N} F_i(t) = (N-1)\sum_{j=1}^{N} F_j(t) - N(\sum_{j=1}^{N} F_j(t) - F_1(t)) \\ &= -\sum_{j=1}^{N} F_j(t) + NF_1(t) = -\sum_{j=2}^{N} F_j(t) + (N-1)F_1(t). \end{aligned}$$

Hence we have $\dot{\eta}(t) = -\eta(t) + F_1(t)$, and $F_s(t) = (N-1)\eta(t) + F_1(t)$ since $F_s(t)$ is the sum of followers' forces and the leader's force. □

We can see that by choosing followers' average force as the state variable, the group force can be put in the format of a standard linear control system $\dot{x} = Ax + Bu,\ y = Cx + Du$. As such, the dynamics from the leader's input force to the resulting group force is first-order, meaning that the leader robot can easily implement feedback control to steer the group force with desired specifications.

Furthermore, the internal state $\eta$ can act as a good approximation of the force of any individual follower robot, as illustrated by the following theorem.

**Theorem 2.** *The difference among all followers in (9) will converge exponentially to zero regardless of the leader's input. The rate of this exponential convergence increases linearly with the number of robots.*

*Proof.* Consider any two follower robots $R_i$ and $R_k$, according to (8)

$$\dot{F}_i(t) - \dot{F}_k(t) = -N(F_i - F_k).$$

So we have

$$F_i(t) - F_k(t) = Ce^{-Nt}. \qquad \square$$

Theorem 2 implies that after a quick transience, all the followers' forces will be the same, so that $\eta$ will be approximately equal to any follower robot's force. Continuing on Theorem 2, we can show that the convergence of the followers' forces to the leader's force is also faster as the number of robots increases. Taking the derivative of $F_s$ in (12) we have $\dot{F}_s = (N-1)\dot{\eta} + \dot{F}_1 = (N-1)(F_1 - \eta) + \dot{F}_1$, where $(F_1 - \eta)$ is the difference between the leader and followers and $(N-1)$ works as the feedback amplifying coefficient for the difference. Therefore the larger $N$ is, the faster $F_s$ will be driven to the desired value.

## 3.3 Controller Design and Trajectory Following

Putting everything together, we can write down the overall state-space dynamics of the system, and derive a controller based on it. The inputs of the system are the leader robot's force $F_1$ and torque $T_1$. The outputs are the object's linear and angular velocity, $v$ and $\omega$.

Choose $v$, $\omega$, $\eta$ as state variables and combine (1),(4) and (12), we get

$$\begin{pmatrix} \dot{\eta} \\ \dot{v} \\ \dot{\omega} \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 \\ \frac{N-1}{M} & -\frac{\mu_v}{M} & 0 \\ 0 & 0 & -\frac{\mu_v}{M} \end{pmatrix} \begin{pmatrix} \eta \\ v \\ \omega \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ \frac{1}{M} & 0 \\ 0 & \frac{1}{J} \end{pmatrix} \begin{pmatrix} F_1 \\ T_1 \end{pmatrix} + \begin{pmatrix} 0 \\ -\mu_s g \frac{v}{\|v\|} \\ 0 \end{pmatrix}. \quad (13)$$

There are a few remarks on the dynamics above: (i) the rotation and translation dynamics are independent from each other, although we write them together for simplicity and clearity; (ii) the nonlinear term induced by the static friction, $-\mu g v_s / \|v_s\|$, can be compensated by offsetting $F_1$, such that the overall dynamics is still linear. Here we briefly show how the compensation works. Let the compensated force $F_1' = F_1 + \mu_s g v/(N\|v\|)$ and divide $\eta$ into two parts: $\eta = \eta_1 + \eta_2$, where $\dot{\eta}_1 = -\eta_1 + F_1$, $\dot{\eta}_2 = -\eta_2 + \mu_s g v/(N\|v\|)$. Note that $\eta_2$ is not affected by $F_1$ so we have $\eta_2 \to \mu_s g v/(N\|v\|)$. Therefore according to (13), $\dot{v} = \frac{N-1}{M}(\eta_1 + \frac{\mu_s g v}{N\|v\|}) - \frac{\mu_v}{M} v + \frac{1}{M}(F_1 + \frac{\mu_s g v}{N\|v\|}) = \frac{N-1}{M}\eta_1 - \frac{\mu_v}{M} v + \frac{1}{M} F_1$, and we can see that the nonlinear term is eliminated.

We use state feedback to achieve the desired system performance. Let $F_1 = K_f(v_d - v) + \mu_s g v/(N\|v\|)$ and $T_1 = K_t(\omega_d - \omega)$, where $v_d$ and $\omega_d$ come from higher-level path planning algorithm. Then the objective is to calculate $K_f$ and $K_t$ according to our specifications. Note that the state feedback only involves proportional control. Integral and derivative control can be implemented by introducing new state variables that are the integral or derivative of the error signal.

Having the controller for $v$ and $\omega$, we can now move on to trajectory following. This requires specifying the desired values of $v_d$ and $\omega_d$. For $v_d$, we use a straightforward vector synthesis strategy. Firstly, we define the point on the desired trajectory that is nearest to object's current position:

$$x_a = \underset{\gamma(\alpha),\alpha\in[0,1]}{\operatorname{argmin}} \|\gamma(\alpha) - x_c\|.$$

Then the desired velocity can be defined as $v_d = w_n v_n + w_t v_t$, as shown in Figure 5, where $v_n = \frac{x_a - x_c}{\|x_a - x_c\|}$, $v_t$ is the unit tangential vector at $x_a$ pointing to the destination and $w_n, w_t$ are just weights. Intuitively, $v_n$ will drag the object towards the trajectory and $v_t$ will maintain the object's velocity along the trajectory. As for rotation, we define $\omega_d = k_\theta(\theta_d - \theta) = k_\theta(g_\theta(x_a) - \theta)$, where $k_\theta$ is a constant gain.

## 4 Simulations

We conduct two manipulation tasks in simulation using Open Dynamic Engine (ODE), a well-known open-source physics engine. The objective of the tasks is to transport an object through an S-shaped maze. In simulation 1, we perform the manipulation for a rectangular plank with 12 robots. In simulation 2, we use 1000 robots to move a large piano of realistic dimensions, which verifies the scalability of our approach.

The snapshots of simulation 1 and 2 are shown in Figure 6 and 7.[3] Both the desired and actual trajectories are shown in Figure 8. The parameters of the environment are: $\mu_s = 0.5$, $\mu_v = 0.3$, $g = 10$. The initial forces of the robots are randomized in the first quadrant, i.e., the angles of the initial forces are in $[0, \frac{\pi}{2}]$. In simulation 1, the force limit of each robot is up to $1.4N$, the torque limit for the leader robot is $5N\cdot m$. In simulation 2, the force limit of each robot is up to $2N$, and the torque limit for the leader robot is $50N\cdot m$. In both simulations, the robotic team successfully transports the object through the maze with rotation being controlled to avoid collision with the wall. Although at the beginning there is a large deviation

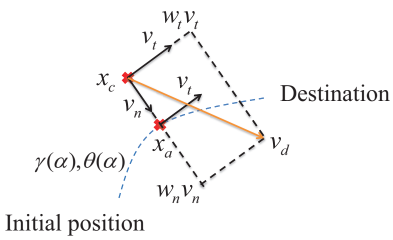


Fig. 5: The synthesis of the desired linear velocity $v_d$.

[3] The video is available online, http://youtu.be/emZVxcl3Zg4

from the desired trajectory due to the random initial motion of the object, the robots can quickly correct the deviation. This is also revealed in Figure 9. Notice that the tracking error goes down when on a straight line and goes up when making a turn. Moreover, the variance of the error of the trajectory following is smaller in 1000 robots case than that in 12 robots case, which verifies that the performance of our algorithm improves as the number of robots increases.

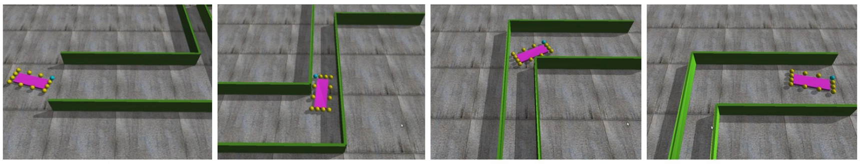

Fig. 6: Simulation 1: manipulation of a small plank (purple) with 12 robots. Dimensions of the object are: weight 1kg, length 0.6m, width 0.2m, height 0.1m. The sphere in blue denotes the leader robot while the follower robots are yellow spheres. The width of the maze varies from 0.5m to 1m.

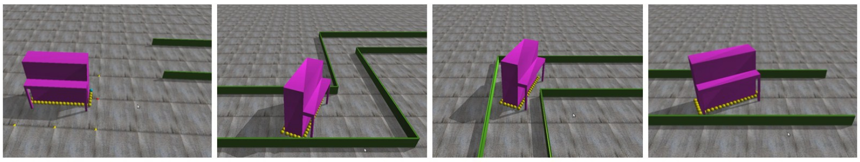

Fig. 7: Simulation 2: manipulation of a large piano (purple) with 1000 robots. The dimensions of the simulated piano are the same as a realistic Steinway K-52 piano: weight 273kg, length 1.54m, width 0.67m, height 1.32m. Robots are centrosymmetrically distributed around the bottom of the piano. For visualization considerations, we draw 40 robots instead of 1000. The width of the maze varies from 1.4m to 2m.

## 5 Conclusion

In this paper, we propose a multi-robot manipulation approach for many small robots to manipulate a massive object. The robots do not have an explicit communication network, but they can locally sense the movement of the object, which gives an indication of the summed forces applied by other robots. We propose a controller by which the robots use the motion of the object itself to reach a consensus on their applied forces. We then design a controller for the leader robot to steer the object based on the analysis of the translational and rotational dynamics of the system. We

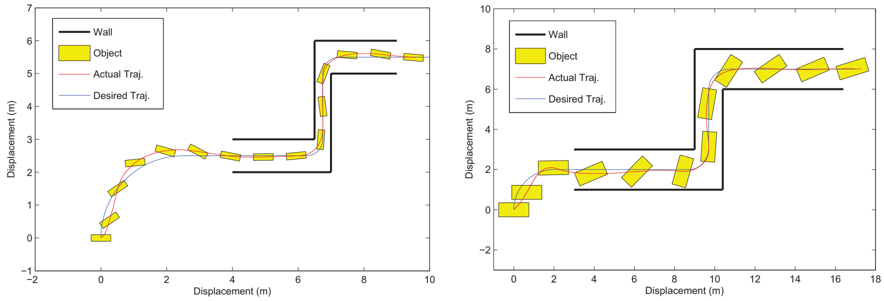


Fig. 8: Overall trajectory of the rectangular plank (left) and the realistic piano (right).

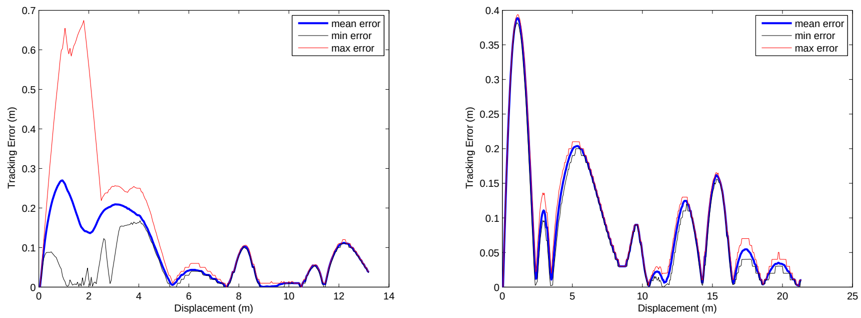


Fig. 9: Trajectory tracking error of consecutive 20 runs for the rectangular plank (left) and the realistic piano (right).

demonstrate the effectiveness of the approach with two simulations implemented in ODE. In the future, we intend to implement our approach experimentally on mobile robot platforms. We are also investigating using parameter adaptation so that robots do not need to know the mass of the object or friction coefficients beforehand, but can learn these quantities on-line.

**Acknowledgements** This work was supported in part by NSF grant CNS-1330036. We are grateful for this support. We also would like to thank James McLurkin and Golnaz Habibi for many insightful discussions on this topic.

## References

1. J. Spletzer, A. Das, R. Fierro, C. Taylor, V. Kumar, and J. Ostrowski. Cooperative localization and control for multi-robot manipulation. In Intelligent Robots and Systems, IEEE/RSJ

International Conference on (2001)
2. Z. Wang and V. Kumar. Object closure and manipulation by multiple cooperating mobile robots. In Robotics and Automation, IEEE International Conference on (2002)
3. G. A. Pereira, M. F. Campos, and V. Kumar. Decentralized algorithms for multi-robot manipulation via caging. The International Journal of Robotics Research, 23(7-8): 783-795 (2004)
4. W. Wan, R. Fukui, M. Shimosaka, T. Sato, and Y. Kuniyoshi. Cooperative manipulation with least number of robots via robust caging. In Advanced Intelligent Mechatronics (AIM), IEEE/ASME International Conference on (2012)
5. Z. Wang, V. Kumar, Y. Hirata, and K. Kosuge. A strategy and a fast testing algorithm for object caging by multiple cooperative robots. In Robotics and Automation, IEEE International Conference on (2003)
6. P. Song and V. Kumar. A potential field based approach to multi-robot manipulation. In Robotics and Automation, IEEE International Conference on (2002)
7. J. Fink, N. Michael, and V. Kumar. Composition of vector fields for multi-robot manipulation via caging. In Robotics: Science and Systems (2007)
8. A. Becker, G. Habibi, J. Werfel, M. Rubenstein, and J. McLurkin. "Massive uniform manipulation: controlling large populations of simple robots with a common input signal." In Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, pp. 520-527 (2013)
9. M. Rubenstein, A. Cabrera, J. Werfel, G. Habibi, J. McLurkin, and R. Nagpal. Collective transport of complex objects by simple robots: theory and experiments. In Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems, pp. 47-54 (2013)
10. R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. Automatic Control, IEEE Transactions on, vol. 49, no. 9, pp. 1520-1533 (2004)
11. A. Jadbabaie, J. Lin, and A. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. Automatic Control, IEEE Transactions on, vol. 48, no.6, pp. 988-1001 (2003)
12. S. Berman, Q. Lindsey, M. S. Sakar, V. Kumar and S. Pratt. Study of group food retrieval by ants as a model for multi-robot collective transport strategies. Robotics: Science and Systems (2010)
13. H. F. McCreery and M. D. Breed. Cooperative transport in ants: a review of proximate mechanisms. Insectes Sociaux (2014)
14. T. J. Czaczkes and F. L. W. Ratnieks. Simple rules result in the adaptive turning of food items to reduce drag during cooperative food transport in the ant Pheidole oxyops. Insectes sociaux (2011)
15. D. Rus, B. Donald, and J. Jennings. Moving furniture with teams of autonomous robots. Intelligent Robots and Systems, IEEE/RSJ International Conference on (1995)
16. K. Böhringer, R. Brown, B. Donald, J. Jennings and D. Rus. Distributed robotic manipulation: Experiments in minimalism. Experimental robotics IV, Springer (1997)
17. B. Donald, L. Gariepy and D. Rus. Distributed manipulation of multiple objects using ropes. Robotics and Automation, IEEE International Conference on (2000)
18. P. Gupta, and P. R. Kumar. The capacity of wireless networks. Information Theory, IEEE International Conference on, vol. 46, no. 2, pp. 388-404 (2000)
19. J. Fink, N. Michael, S. Kim, and V. Kumar. Planning and control for cooperative manipulation and transportation with aerial robots. The International Journal of Robotics Research, vol. 30, no. 3, pp. 324-334 (2011)
20. M. Ji, A. Muhammad, and M. Egerstedt. Leader-based multi-agent coordination: Controllability and optimal control. American Control Conference, pp. 1358-1363 (2006)
21. H. G. Tanner, G. J. Pappas and V. Kumar. Leader-to-Formation Stability. Robotics and Automation, IEEE Transactions on, vol. 20, no. 3, pp. 443-455 (2004)
22. G. Habibi, L. Schmidt, M. Jellins, J. McLurkin. K-redundant trees for safe multi-robot recovery in complex environments. International Symposium on Robotics Research (2013)