

A Decentralized, Communication-Free Force Distribution method with application to Collective Object Manipulation

Shadi T. Kalat
Mechanical Engineering Department,
Worcester Polytechnic Institute,
Worcester, MA
stasdhikalat@wpi.edu

Siamak G. Faal
Soft Robotics Laboratory,
Robotics Engineering,
Worcester Polytechnic Institute,
Worcester, MA
sghorbanifaal@wpi.edu

Cagdas D. Onal*
Soft Robotics Laboratory,
Mechanical Engineering Department,
Worcester Polytechnic Institute,
Worcester, MA
cdonal@wpi.edu

We present a novel approach to achieve decentralized distribution of forces in a multi-robot system. In this approach, each robot in the group relies on the behavior of a cooperative virtual teammate that is defined independent of the population and formation of the real team. Consequently, such formulation eliminates the need for inter-agent communications or leader-follower architectures. In particular, effectiveness of the method is studied in a collective manipulation problem where the objective is to control the position and orientation of a body in time. To experimentally validate the performance of the proposed method, a new swarm agent, $\Delta\rho$ (Delta-Rho) is introduced. A multi-robot system, consisting of five $\Delta\rho$ agents is then utilized as the experimental setup. The obtained results are also compared with a norm-optimal centralized controller by quantitative metrics. Experimental results prove the performance of the algorithm in different tested scenarios and demonstrate a scalable, versatile, and robust system-level behavior.

1 Introduction

Swarm systems demonstrate global intelligent behavior emerging from local interactions among many simple agents. One of the frequently observed behaviors in biological swarm systems is the collective food retrieval, in which multiple insects (e.g. *Eciton Burchellii* also known as *Eciton* army ants) carry a relatively large prey to their nest. This observations inspired many researchers [1, 2] to design novel robotic systems [3]. Unlike frameworks that

focus on increasing dexterity and intelligence of a single well-instrumented robot, swarm systems utilize many low-cost, simple robots to realize complex tasks. Such systems demonstrate a scalable, flexible, and robust behavior [4] as a result of being spatially distributed, following simple rules and having no single point of failure.

This article introduces a decentralized force control algorithm and analyzes its application in collective object manipulation. The proposed algorithm substitutes the real (and unknown) group formation with a hypothetical (and known) formation, which is composed of an agent and its virtual teammate. Since a single virtual agent can also represent the effect of assuming multiple virtual agents, this number would appear as a scaling factor in the force distribution. Consequently, coordination between each agent and its virtual teammate produces the collective manipulation behavior.

The presented results prove that this imprecise agent-level assumption yields to successful pose control of the manipulated object without requiring any inter-agent communication or leader-follower architecture. Additionally, the proposed method has a number of advantages over current methods. These advantages include: 1) The implementation of the algorithm does not require any information about the population and formation of the group. Consequently, none of the agents needs to exchange information with other group members; 2) Coordination between the agents is achieved without relying on a group leader which increases system robustness; and 3) Modulating local forces exerted on the object (instead of planning paths and controlling positions of agents) gener-

*Corresponding author

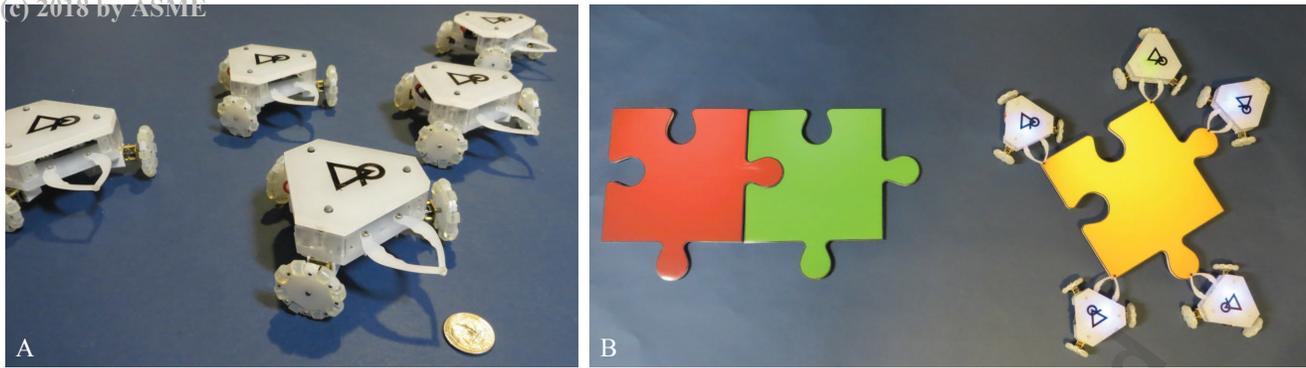


Fig. 1. (A) $\Delta\rho$ swarm agents. (B) A multi-robot system consisting of five $\Delta\rho$ robots as they are manipulating a puzzle piece

ates a degree of mechanical compliance in the overall system behavior. The proposed algorithm is a fundamental framework that is open to extension with the implementation of impedance or force control of the manipulated object considering its interaction with the environment.

Experimental validation of the performance of the proposed algorithm inspired the design and fabrication of a new robotic platform: $\Delta\rho$. Due to its holonomic locomotion system, $\Delta\rho$ is capable of moving and applying forces in any planar direction. $\Delta\rho$ is fabricated from interlocking 2-D profiles created by laser machining, which significantly reduces the fabrication cost and time. Fig. 1-A illustrates the $\Delta\rho$ platforms and Fig. 1-B presents a snapshot of 5 $\Delta\rho$ robots as they are manipulating a puzzle piece to a desired pose.

Stability and convergence of the proposed controller are analyzed in detail and its scalability, versatility, and robustness are validated through a set of experiments with a custom multi-robot system. Efficiency and effectiveness of the algorithm are evaluated in a number of experimental scenarios by quantitative metrics including manipulation time, path efficiency, and velocity profile variations. The calculated metric values are then compared with a force-optimal centralized controller [5], where all agents are aware of the group population and formation. Theoretical analyses and experimental results indicate that the presented collective manipulation algorithm offers significant potential for the future of swarm robotics, towards realizing large-scale, communication-free swarm missions with applications in search-and-rescue, construction, and warehouse automation.

The rest of the article is organized as follows. Section 1.1 presents a brief background and available methods in the literature that tackle the collective manipulation problem. Section 2 discusses the mathematical modeling and derivation of the decentralized controller based on a set of assumptions presented in Section 1.2. Design, fabrication, and control of the $\Delta\rho$ robots are discussed in Section 3. Section 4 covers the details about instrumentation and data acquisition system, and presents the results of several experiments conducted to evaluate the effectiveness and efficiency of the proposed method. The article is concluded with discussions and future work in Section 5.

1.1 Related work

In general, several factors could determine the complexity and effectiveness of collective behaviors in swarm systems. In the case of collective manipulation, these factors include shape and size of the object, population of the group, amount of information available to the agents, and their physical capabilities. These capabilities determine whether the object can be manipulated using force or form closure methods.

Some known challenges associated with force closure are stagnation, coordination of motion, and the effect of the shape of the transported object. One of the earlier works, [6] addresses the issue of stagnation and proposes a recovery mechanism, which utilizes the application of random forces by either realigning the direction of forces or repositioning the pushing force.

In form closure approaches, [7] addresses some of the associated problems by calculating the minimum population and the group formation which ensures that the relative degree of freedom of the object is zero. A decentralized approach for confining an object with multiple mobile robots is studied in [8]. Their proposed algorithm is based on a gradient descent method for a system with a known object shape and known relative positions of the agents with respect to the object. The task of capturing a target is divided into two sub-tasks of enclosing and grasping. The objective function is defined to uniformly distribute the agents around the object by minimizing the angular distance between an agent and its neighbors around the object.

Regardless of the strategy used for collective manipulation, successful force/motion coordination requires an agreement on the goal position if the goal is not visible to all agents. This agreement can be obtained either by propagating the goal location to all agents or relying on group leaders, which are assumed to know the goal location. This approach requires a consensus strategy in the team as described in [9, 10]. Based on this method, agents which are aware of the goal location will move towards it, while other agents try to minimize their heading error with respect to their neighbors. Another consensus approach is studied in [11] where some robots are aware of the location of the target and the others (referred to as blind agents) are not. In this method,

the robots that are connected to the object are treated as parts of the object; thus allowing the other agents to attach to them. If a robot is not blind, it can simply head towards the goal location and set its speed to the maximum value. Blind robots can perceive forces and consequently determine the desired orientation of and move in that direction.

Consensus can also be attained via physical feedback from the object. This approach is studied in [12, 13], where force feedback from the object is used to adjust the direction of forces applied by the follower agents to follow the direction of a leader. Although consensus based algorithms work well in coordinating the movement of agents, the implementation of such methods on robotic platforms requires communication channels or sensory equipment, which may reduce the applicability of the methods in real life applications. In addition, the heterogeneous nature of leader-follower schemes decrease the robustness of the system as the group members rely on certain individuals to coordinate their motion. In contrast to having partial goal visibility, some studies [6, 14] assume that the goal point is visible to all the agents. Therefore, robots only need to move the object in a defined direction, while the orientation of the object is not controlled.

In addition, current approaches demand at least one type of information exchange between robotic agents to provide a successful coordination [15] or full control on position and orientation of the object throughout the manipulation process. This exchange of information can be either in a direct communication form, which can affect the scalability of the system, or physical feedback such as forces or motion, which may reduce system robustness. In this regard, [16] proposed an algorithm to reduce the communication requirements of a swarm system. Their proposed algorithm is based on position control of robots, where a global error signal is sent for all agents to regulate. Although this algorithm was able to manipulate objects to a desired position and orientation, it is subject to various limitations. The first drawback of the approach is that the system is not controllable in an obstacle-free workspace due to the reduced rank of the global controllability matrix. The other drawbacks include time complexity, scalability, and the inability to keep force or form closure around the object while following a trajectory.

1.2 Assumptions

In contrast to the related work, our algorithm does not explicitly account for closure around the object, but relies on robots being rigidly connected to the object with a non-prehensile end-effector that allows the robot to rotate freely around the connection point. The robots do not know the shape or mass of the object, or the group population or configuration. We do not utilize a leader-follower architecture, but assume that all agents are aware of the target position and orientation (pose) of the object. In addition, we assume that each agent can monitor its own pose and the pose of the object. Experimentally, we use a motion capture system to relay this information to each agent.

2 Problem Formulation

Although the presented algorithm is extensible to 3-D space, this manuscript focuses on the formulation and experimental validation of our method for a multi-robot system constrained to planar motion. The vector and coordinate frame notations used in this article are adopted from [17]. Based on these notations, a transformation A from coordinate frame $\{i\}$ to coordinate frame $\{j\}$ is denoted by jA_i . Similarly, ${}^i v$ illustrates vector v defined in coordinate frame $\{i\}$. It is assumed that the coordinate frame $\{O\}$ is attached to the object center of mass and the vector ${}^O r_i \in \mathbb{R}^2$ defines the attachment point of agent i measured from the object center of mass in $\{O\}$. Assuming that the robots are only able to apply forces but not moments, vector ${}^O f_i \in \mathbb{R}^2$ defines the force applied by the i^{th} agent in coordinate frame $\{O\}$. The error vector, ${}^O e \in \mathbb{R}^3$, is composed of linear and angular differences between desired and current object positions in $\{O\}$ and defined as ${}^O e = (x_d - x_o) {}^O \hat{i} + (y_d - y_o) {}^O \hat{j} + (\theta_d - \theta_o) {}^O \hat{k}$. Equation (1) describes the differential equations of motion for an object with mass $m \in \mathbb{R}$ and mass moment of inertia $I \in \mathbb{R}$ in the body-fixed (non-inertial) reference frame $\{O\}$. The vectors ${}^O v_o \in \mathbb{R}^2$ and ${}^O f_i \in \mathbb{R}^2$ represent the velocity of the object and the forces applied by each agent, respectively.

$$\begin{bmatrix} m({}^O \dot{v}_o + \dot{\theta}_o {}^O \hat{k} \times {}^O v_o) \\ I \ddot{\theta}_o {}^O \hat{k} \end{bmatrix} = \sum_{i=1}^N \begin{bmatrix} {}^O f_i \\ {}^O r_i \times {}^O f_i \end{bmatrix}. \quad (1)$$

In the above equation, $N \in \mathbb{N}$ represents the total number of agents that are involved in the task. Unless noted otherwise, throughout the rest of this article, all the vectors are defined in coordinate frame $\{O\}$. Thus, for the sake of brevity, the superscript O is dropped from vector names in the following sections. The right side of Eqn.(1) can be reformulated as:

$$\sum_{i=1}^N \begin{bmatrix} f_i \\ r_i \times f_i \end{bmatrix} = J [f_1^T, f_2^T, \dots, f_N^T]^T = F, \quad (2)$$

where the matrix $J \in \mathbb{R}^{3 \times 2N}$ is the full Jacobian of the system [17, 18] (note that, in some context, based on the definition of the joint and work spaces, the matrix J is referred to as the grasp matrix [19]). In this equation $r_{xi} \in \mathbb{R}$ and $r_{yi} \in \mathbb{R}$ are the x and y components of the r_i vector respectively. Full system Jacobian is written explicitly as:

$$J = \begin{bmatrix} 1 & 0 & 1 & 0 & \dots & 1 & 0 \\ 0 & 1 & 0 & 1 & \dots & 0 & 1 \\ -r_{y1} & r_{x1} & -r_{y2} & r_{x2} & \dots & -r_{yN} & r_{xN} \end{bmatrix}. \quad (3)$$

Derivation of the decentralized controller is inspired from a centralized controller that utilizes the full Jacobian of the system and Moore-Penrose pseudoinverse [20] to effectively distribute the control action among the team agents. The details on formulation of the decentralized controller are

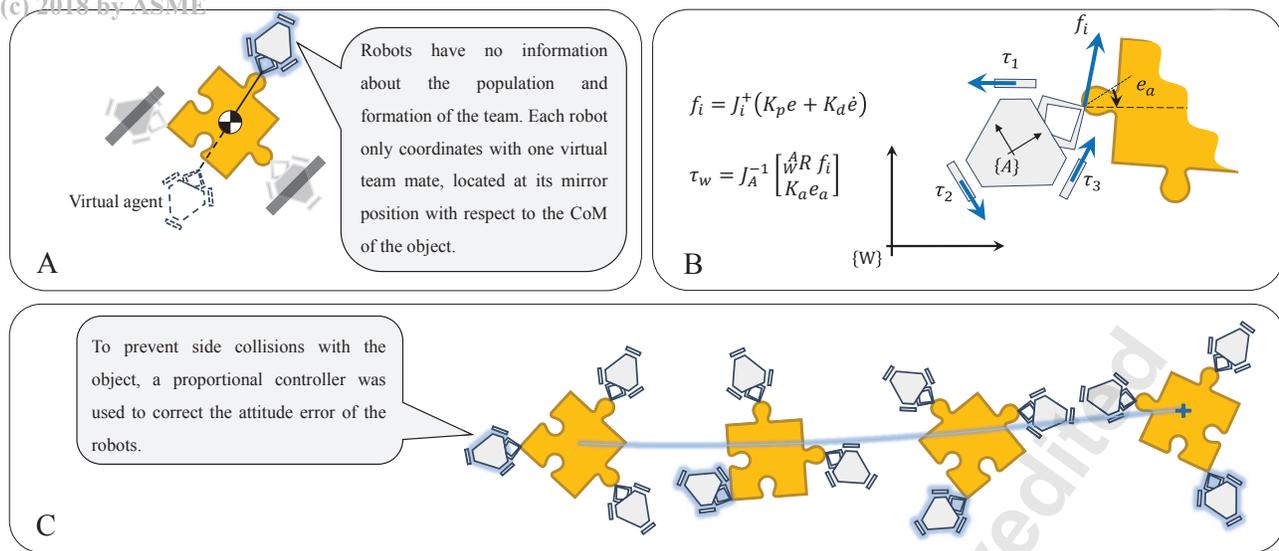


Fig. 2. (A) System virtual configuration from each robot's view. The virtual configuration for each robot consists of the robot itself and a virtual teammate located at its mirror location with respect to the object center of mass. (B) Utilization of the virtual agents eliminates the need for inter-agent communications and enables each robot to calculate its force based on the error vector of the object. The traction forces of the wheels are then computed using the Jacobian of the robotic platform, J_A . (C) Graphical illustration of the experiment with 200 g payload. As observed from the figure, robots successfully move the object to the desired position and orientation as they are minimizing their attitude error e_a with respect to the object

discussed in what follows. Throughout the rest of this article the terms “pseudoinverse” and “Moore-Penrose pseudoinverse” are used interchangeably.

can assume the following local Jacobian matrix $J_i \in \mathbb{R}^{3 \times 4}$.

$$J_i = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ -r_{yi} & r_{xi} & r_{yi} & -r_{xi} \end{bmatrix}. \quad (4)$$

2.1 Decentralized Algorithm

In the derivation of the decentralized controller, it is assumed that each agent only cooperates with a virtual agent that is located at an arbitrary position around the object. If the virtual agent simulates the effect of the rest of the team, the response of the decentralized system will converge to the centralized approach. We presented the details of derivation, convergence proofs, and numerical validation of this claim in [5] over simulation results. As noted in [5], although virtual agents may not necessarily represent the effect of the rest of the team, they provide a means to distribute the force vector F among the agents. Thus, the problem reduces to defining the positions of virtual agents such that the transformation $T : \phi(e) \mapsto F$ remains positive definite. A possible solution that guarantees positive definiteness of T for a physical system (agents with finite dimensions) is to define the location of the virtual agents at the mirror positions of the team members with respect to the object center of mass as shown in Fig.2-A. Note that the mirror position is desirable because it significantly simplifies the derivations and it can be replaced with any other position as long as it results in a positive definite T matrix. Using this approach, each agent

The manipulation problem is formulated such that each agent only knows a minimal set of information: the point of attachment to the object r_i , and the error vector e [12]. Thus, employing J_i in deriving the local forces eliminates inquiring information about the real team composition. Finally, by substituting the feedback control law $\phi(e)$ with a PD control function, the complete decentralized control law for each agent can be written as:

$$f_i = J_i^+ \phi(e) = J_i^+ (K_p e + K_d \dot{e}), \quad (5)$$

where J_i^+ is the Moore-Penrose pseudoinverse of the local Jacobian matrix J_i as shown in Fig. 2-B. This formulation results in the required planar forces by each agent to collectively manipulate the object to a desired location and orientation. These forces are defined in the object coordinate frame $\{O\}$ without considering geometries of the robots surrounding the object.

Imposed physical constraints from a real-world implementation of the algorithm demands local control on the orientation of the robots to eliminate possible collisions. These constraints include curvature, concavity and convexity of the object shape at the points of attachment, object dimensions,

and population of the team, which affect the range of attainable angles of attachment. Thus, to avoid collisions, in addition to providing the desired forces, robots need to control their orientations with respect to the object.

The controller formulation above determines linear planar force vectors at the attachment points to the object. Since applying a planar force vector requires only two degrees of freedom, the extra degree of freedom provided by the holonomic locomotion system of $\Delta\rho$ is utilized to control the relative angle between the robot and the object as a secondary goal as described in Section 3. The effect of the controller used to correct the orientation of the robots with respect to the object is depicted in Fig. 2-C extracted from experimental data, where agents work to remain in a constant relative orientation with respect to the object through the course of manipulation.

2.2 Stability and Convergence Analysis

The objective is to define all the components of f_i such that the norm of the error vector converges to $B(\epsilon)$, $\epsilon > 0$, in finite time; where $B(\epsilon) = \{e \in \mathbb{R}^3 : \|e\| < \epsilon\}$. This can be achieved by equating $F = [F_x, F_y, M_z]^T$ to the output of a control function, $\phi(e)$. Although any linear or nonlinear controller can be utilized as the control function, considering dynamics of the system, $\phi(e)$ is simply set to be a proportional-derivative (PD) controller. The output of the control function $\phi(e)$ can be mapped into agent forces by solving Eqn.(2) for f_i . Since J is not a square matrix, there is no unique solution for the corresponding system of equations. Thus, Moore-Penrose pseudoinverse is utilized to obtain a minimum Euclidean norm solution for f_i . The complete centralized controller is formulated as:

$$[f_1^T, \dots, f_N^T]^T = J^+ \phi(e) = J^+ (K_p e + K_d \dot{e}), \quad (6)$$

where J^+ is the Moore-Penrose pseudoinverse of the Jacobian matrix J . The resultant applied force to the object center of mass for each controller can be obtained by substituting the forces obtained using the corresponding control equations into Eqn.(2). For the case of the centralized approach this substitution yields:

$$\sum_{i=1}^N \begin{bmatrix} f_i \\ r_i \times f_i \end{bmatrix} = J [f_1^T \dots f_N^T]^T = J [J^+ \phi(e)]. \quad (7)$$

As observed in Eqn.(3), the rows of the Jacobian matrix J are linearly independent. Thus multiplication of J by its pseudo-inverse results in an identity matrix and Eqn.(7) reduces to:

$$\sum_{i=1}^N \begin{bmatrix} f_i \\ r_i \times f_i \end{bmatrix} = \phi(e). \quad (8)$$

Consequently, convergence and stability of the centralized controller directly depends on the behavior of the control

function $\phi(e)$. If $\phi(e)$ guarantees system stability, the centralized controller will also be stable. Moreover, the system will demonstrate the same response as if it is directly controlled by $\phi(e)$.

Following a similar approach, we can find the total force applied to the object center of mass for the decentralized controller by substituting Eqn. (5) into Eqn.(2) that leads to

$$\sum_{i=1}^N \begin{bmatrix} f_i \\ r_i \times f_i \end{bmatrix} = J [f_1^T, \dots, f_N^T]^T = J [K_1, \dots, K_N]^T \phi(e), \quad (9)$$

where K_i is the first two rows of J_i^+ and defined as:

$$K_i = \frac{1}{2\mathfrak{R}_i} \begin{bmatrix} k_{i11} & k_{i12} & k_{i13} \\ k_{i21} & k_{i22} & k_{i23} \end{bmatrix}, \quad (10)$$

$$\begin{aligned} \mathfrak{R}_i &= (a_{xi} - r_{xi})^2 + (a_{yi} - r_{yi})^2, \\ k_{i11} &= \mathfrak{R}_i + a_{yi}^2 - r_{yi}^2, \\ k_{i12} &= -(a_{xi} + r_{xi})(a_{yi} - r_{yi}), \\ k_{i13} &= 2(a_{yi} - r_{yi}), \\ k_{i21} &= -(a_{xi} - r_{xi})(a_{yi} + r_{yi}), \\ k_{i22} &= \mathfrak{R}_i + a_{xi}^2 - r_{xi}^2, \\ k_{i23} &= -2(a_{xi} - r_{xi}). \end{aligned}$$

In the above equation, $a_i = [a_{xi}, a_{yi}]$ is the location of the virtual agent i that is defined in the object coordinate frame $\{O\}$. In general, $[K_1, K_2, \dots, K_N]^T$ will not be equal to the pseudoinverse of J and the right hand side of Eqn.(9) will not simply reduce to $\phi(e)$. Consequently, the behavior of the control function $\phi(e)$ will be affected by the nature of the resultant transformation matrix T . This transformation matrix, which maps the output of the control function to the forces that are applied to the object center of mass, is defined as:

$$T = J [K_1, K_2, \dots, K_N]^T. \quad (11)$$

If T is positive definite, the inner product between the resultant transformed control actions and the vector $\phi(e)$ will be positive. Thus, a positive definite matrix T preserves the behavior of $\phi(e)$ and results in a stable mapping between the control function and the forces applied to the object center of mass.

Although a_i vectors can have any arbitrary values as long as they yield to a valid T matrix, the formulation presented in this article assumes that the virtual agent for i^{th} agent is located at its mirror position with respect to the center of mass of the object. Substituting values of $a_i = -r_i$ into Eqn.(10) yields:

$$K_i = \frac{1}{2} \begin{bmatrix} 1 & 0 & -r_{yi}/(r_{xi}^2 + r_{yi}^2) \\ 0 & 1 & r_{xi}/(r_{xi}^2 + r_{yi}^2) \end{bmatrix}. \quad (12)$$

Finally, substituting K_i values into Eqn.(11) yields to

the transformation matrix T of proposed decentralized algorithm:

$$T = \frac{1}{2} \begin{bmatrix} N & 0 & -\sum_{i=1}^N r_{yi}/(r_{xi}^2 + r_{yi}^2) \\ 0 & N & \sum_{i=1}^N r_{xi}/(r_{xi}^2 + r_{yi}^2) \\ -\sum_{i=1}^N r_{yi} & \sum_{i=1}^N r_{xi} & N \end{bmatrix}, \quad (13)$$

and the corresponding eigenvalues of the matrix T are $\lambda = \{N - \delta, N, N + \delta\}$, where δ is defined as

$$\delta = \sqrt{\sum_{i=1}^N \frac{r_{yi}}{r_{xi}^2 + r_{yi}^2} \sum_{i=1}^N r_{yi} + \sum_{i=1}^N \frac{r_{xi}}{r_{xi}^2 + r_{yi}^2} \sum_{i=1}^N r_{xi}}. \quad (14)$$

All the eigenvalues of T are positive if and only if $-N < \delta < N$. Since the perimeter of the object is finite and bounded, and the physical agents have a finite and nonzero perimeter, the growth in population of the group causes the matrix T to approach to a scaled identity matrix. As a result, all the eigenvalues of T remain greater than zero and a stable mapping is obtained between $\phi(e)$ and forces applied to the object center of mass. Moreover, the eigenvalues of T are scaled by the population of the team, N . Since the Euclidean norm of the real matrix T is equal to square root of its maximum eigenvalue ($\|T\| = \sqrt{N + \delta}$), the decentralized approach results in larger forces applied to the object center of mass and consequently a shorter settling time. Also, since the condition number of the T matrix is ratio between its maximum and minimum eigenvalues ($\kappa(T) = (N + \delta)/(N - \delta)$), as N approaches to infinity, $\kappa(T)$ approaches to 1. Thus, the system performance converges to the centralized approach for highly populated groups.

In the formation experiments presented in this article, the robots are located around a circular object. Using this information, it is possible to further simplify the expression of δ by substituting $r_{xi} = R \cos(\theta_i)$ and $r_{yi} = R \sin(\theta_i)$ in Eqn.(14). For an object with a fixed radius R , θ_i defines the angle for the attachment point of the i^{th} agent. Finally, a simpler expression for δ is obtained as:

$$\delta = \sqrt{\left(\sum_{i=1}^N \sin(\theta_i)\right)^2 + \left(\sum_{i=1}^N \cos(\theta_i)\right)^2}. \quad (15)$$

Thus, as the agents get closer to each other, the value of δ will increase which results in a larger norm and condition number of T . Consequently, the system will have a faster response with a larger steady-state error.

3 Delta-Rho Robotic Platform

The large population of agents involved in a swarm system demands low-cost platforms to serve as team members.

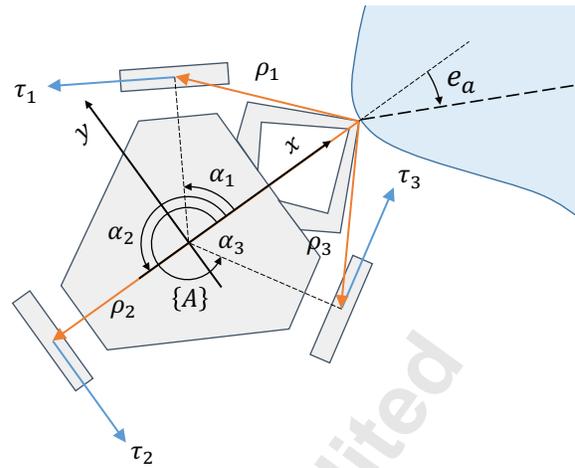


Fig. 3. Description of the parameters used for derivation of the control equations for robots

Consequently most of the potential robotic swarm agents introduced so far utilize a simple locomotion system. Some of the common approaches for locomotion system designs includes: vibration based locomotion [21], differential drives [22], and two and three degrees of freedom legged locomotion systems [23–25]. Although our earlier work [24] utilized a legged holonomic structure, the legged nature of its locomotion system cannot provide a continuous force output capability. To simplify the experimental setup and to focus on the performance of the algorithm, it is desirable to use a platform with a continuous force control capability without the rocking motions caused by discrete phases of legged locomotion. $\Delta\rho$ is a small, accessible holonomic drive robot capable of applying forces in any arbitrary planar direction parallel to the substrate surface using a non-prehensile end-effector without utilizing any active arm mechanisms. The design, control and fabrication details of the $\Delta\rho$ robotic platform are discussed in what follows.

3.1 Robot Design and Control

$\Delta\rho$ is a holonomic mobile robot, which is specifically designed as a testbed for multi-robot and swarm algorithms. A 3-wheel holonomic platform serves as the locomotion system of the robot which enables motions in any arbitrary planar direction. Therefore, $\Delta\rho$ is capable of applying forces in any planar direction without utilizing an active arm. Each robot is identified by a number of infrared reflective markers that are placed on the top surface of the body. Although the platform can be equipped with on-board localization sensors, to eliminate the errors associated with position estimations, position and orientation of the robots are directly tracked with an OptiTrack motion capture system.

In what follows, we describe the control formulation for each robot. The parameters associated with $\Delta\rho$ agents are depicted in Fig. 3. The traction forces of the robot wheels, $\tau_w = [\tau_1, \tau_2, \tau_3]^T \in \mathbb{R}^3$, is related to force and moment vec-



Fig. 4. An overview of the experimental setup used for validation of the proposed algorithm

tors at the tip of the end effector, $[T_x, T_y, C_z]^T \in \mathbb{R}^3$, by

$$\begin{bmatrix} T_x \\ T_y \\ C_z \end{bmatrix} = \sum_{w=1}^3 \underbrace{\begin{bmatrix} -\sin(\alpha_w) \\ \cos(\alpha_w) \\ \rho_{wx} \cos(\alpha_w) - \rho_{wy} \sin(\alpha_w) \end{bmatrix}}_{J_R} \tau_w, \quad (16)$$

where $\alpha_w \in \{\pi/3, \pi, -\pi/3\}$ is the angle between wheel w and positive A_x axes. Variables ρ_{wx} and ρ_{wy} in \mathbb{R} are the x and y components of ρ vector measured from the tip of the end effector to the center of w and $J_R \in \mathbb{R}^{3 \times 3}$ represents the robot Jacobian. Finally, the traction force for each wheel of the robot is calculated by substituting Eqn.(5) for T_x and T_y and solving $\tau_w = J_R^{-1} [T_x, T_y, C_z]^T$. To correct the attitude of the robot with respect to the object, moment C_z is calculated by applying a proportional controller with gain $K_a < 0 \in \mathbb{R}$ to the attitude error, $e_a \in \mathbb{R}$. Therefore, τ_w is defined as

$$\tau_w = J_R^{-1} \begin{bmatrix} {}^A R [\Omega (K_p e + K_d \dot{e})] \\ K_a e_a \end{bmatrix}, \quad (17)$$

where Ω is a matrix composed of the first two rows of J_i^+ . The above equation determines the required traction forces for each wheel of the robot for a given manipulation force and attitude error. The traction force for each wheel is then converted to the required motor torque based on the gear ratio and diameter of the wheel. For each robot, linear velocity of the wheel centers is computed using J_R^T . Consequently, motor torques are controlled with an open-loop controller based on the motor parameters. For a fully contained solution, one can also use the closed-loop torque control based on motor current feedback as discussed in [26].

3.2 Fabricated Prototype

The robot structure and wheels are fabricated by CO₂ laser machining and assembly of interlocking 2-D profiles that are cut from 2 mm and 6 mm thick acrylic sheets. On-board actuators and sensors are controlled by a custom con-

trol board that utilizes an Atmel ATmega1284P microcontroller. Two DRV8833 Dual H-Bridge motor drivers are used to control the input voltage of the three permanent-magnet DC (PMDC) motors that drive the three holonomic wheels of the robot. An XBee RF transmitter is connected to the main control board to allow communication with external devices. Each of the robots use one 7.4 V 180 mAh 2-cell lithium polymer battery as their power source. Five fabricated prototypes are shown in Fig. 1. Each robot weighs 150 g and fits into a 127×117×50 mm box.

3.3 Experimental setup

The experimental setup consists of up to five $\Delta\rho$ robots, which manipulate objects of various weights in the horizontal plane. To focus the experiments on the performance of the algorithm, it is assumed that grasping of the object has already been achieved by the robots. Therefore, finding and attaching to the object are not discussed in this work. To ensure a robust physical connection between agents and the object through the course of experiments, $\Delta\rho$ robots are attached to the object by pin joints. Moreover, to eliminate the errors associated with pose estimation, an Optitrack motion capture system with four cameras (18 μ m accuracy) is used to detect the position and orientation of the robots and the object. This information is processed in MATLAB and sent to the robots over an XBee network. An overview of the experimental setup is illustrated in Fig. 4. The experiments are conducted with two circular objects with masses of 100 and 230 g. The average coefficients of static and kinetic friction measured in different locations of the experimental environment are 0.44 ± 0.1 and 0.22 ± 0.07 , respectively.

4 Results and Discussions

One of the most important features of a swarm system is to demonstrate scalable, flexible, and robust system-level functionality [4]. Thus, several experimental scenarios are designed to study the system-level behavior and evaluate the efficiency of the proposed algorithm in different conditions. The experimental scenarios were tested with both decentralized and centralized controllers and responses of both control strategies are compared and tabulated.

Figure 5 illustrates snapshots of three $\Delta\rho$ robots as they carry a 100 g object to a desired pose, depicted with dashed white line. The complete explanation of the algorithm, robot control method and details about $\Delta\rho$ platforms are also presented in [27].

4.1 Scalability

Scalability requires the system to be able to operate with different group populations. To study this property, we varied the number of agents for manipulating a 230 g object between fixed start and goal poses. It can be observed from Fig. 6, the proposed algorithm is able to find similar solutions with different group populations.

The results of this experiment suggest that increasing the group population will result in smaller position and orienta-

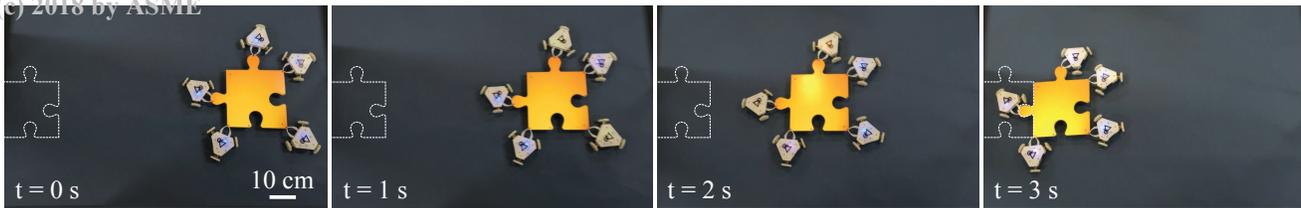


Fig. 5. Snapshots of five Δp robots as they are manipulating a 100 g object. The manipulated object is assembled to a virtual puzzle piece depicted with white dashed line

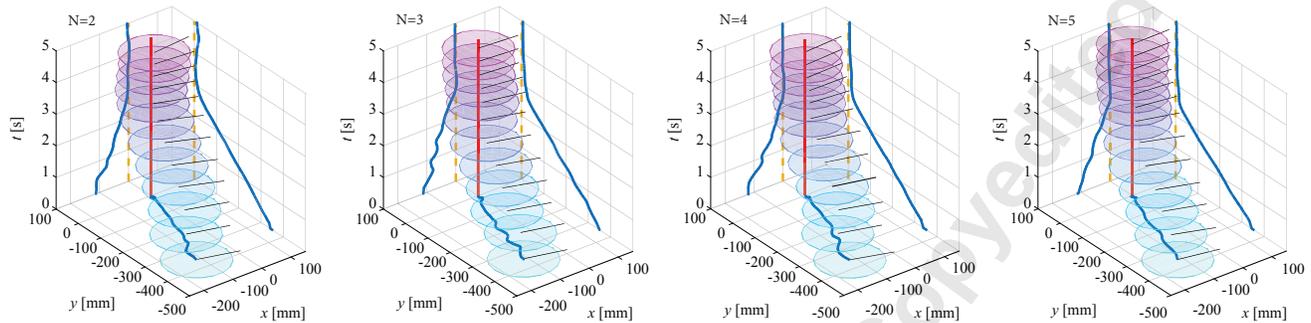


Fig. 6. Position and orientation of the object over time (vertical axis) for different group populations N . Path of the object center of mass is projected on x - y plane (depicted with blue color). Time responses for x and y are plotted in y - t and x - t planes respectively. The dashed yellow line on x - t and y - t planes illustrate the desired positions along x and y axes, respectively. The red line represents the location of the goal over time on the x - y plane. The orientation of the object over time is depicted via the black line parallel to the x axis of the object. The color gradient of the object, varying from cyan to magenta, illustrates the passage of time.

tion errors and shorter settling times for the system. This is expected due to an increase in the resultant forces applied to the object center of mass based on the model presented in Eqn.(1). This increase in total applied force allows the system to readily overcome static friction and consequently reach closer to the goal position. It is also observed that the settling time for the decentralized controller is smaller than the centralized controller. This is due to the assumption we made on the existence of only one virtual agent for each real agent, which yields larger forces applied by the group. This phenomenon is proved by noting that the eigenvalues of matrix T in Eqn. (13) in Section 2.2 are greater than the eigenvalues of the identity matrix $J[J^+]$ (which is equal to 1) for the centralized controller.

Table 1 presents the average settling times for manipulating the object. Throughout this article, two settling time metrics are defined as the time elapsed from the start of the experiment to the time at which the object center of mass enters and remains within 15% and 5% error bands, respectively. The average steady-state error for centralized and decentralized controllers are also presented in Table 1.

In all trials of this experiment, the robots are distributed uniformly around the object with 72° increments. Such formations eliminate any overlap in the locations of virtual agents with real robots. Therefore, the results presented in Fig. 6 and Table 1 essentially correspond to worst case scenarios of the decentralized controller due to the fact that none of the virtual agents exactly overlaps with a real robot. We

Table 1. Settling time and steady-state error for different group populations N

| Settling time | | | | | |
|--------------------|--|-------------------|---|-------------------|--|
| | | $e = 15\%$ | | $e = 5\%$ | |
| N | Dec. | Cent. | Dec. | Cent. | |
| 2 | 2.31 ± 0.40 s | 2.72 ± 0.48 s | 2.79 ± 0.66 s | 3.32 ± 0.64 s | |
| 3 | 2.25 ± 0.32 s | 2.39 ± 0.44 s | 2.70 ± 0.80 s | 3.03 ± 0.52 s | |
| 4 | 2.17 ± 0.16 s | 2.18 ± 0.24 s | 2.57 ± 0.48 s | 2.94 ± 0.34 s | |
| 5 | 1.91 ± 0.24 s | 2.12 ± 0.30 s | 2.28 ± 0.38 s | 2.78 ± 0.40 s | |
| Steady-state error | | | | | |
| N | Dec. | | Cent. | | |
| 2 | 9.1 ± 0.97 mm, $11.4 \pm 2.40^\circ$ | | 7.1 ± 0.81 mm, $8.8 \pm 1.92^\circ$ | | |
| 3 | 4.3 ± 0.66 mm, $5.7 \pm 1.21^\circ$ | | 3.5 ± 0.75 mm, $4.3 \pm 0.85^\circ$ | | |
| 4 | 3 ± 0.78 mm, $2.2 \pm 0.93^\circ$ | | 2.4 ± 0.74 mm, $1.4 \pm 0.68^\circ$ | | |
| 5 | 0.8 ± 0.41 mm, $0.6 \pm 0.47^\circ$ | | 0.2 ± 0.30 mm, $0.2 \pm 0.11^\circ$ | | |

previously studied this effect in [5] over a set of simulations. In real applications of the algorithm, it is more likely to have overlaps in more populated swarm systems or even-numbered groups with a uniform distribution of agents.

Although overlap of the virtual and real agents will not affect the settling time of the system, it will reduce the steady-state error. As an example, in an experiment con-

ducted with four uniformly distributed agents (90° increments), the steady-state error was reduced to 0.4 mm and 1.9° . In general, the steady-state error for the centralized controller is smaller than the decentralized one, but as the group population increases, the decentralized controller response approaches that of the centralized method.

4.2 Versatility

Versatility of the system is defined as the ability to find a feasible solution to the collective manipulation task in response to changes in experimental conditions or the environment. In the current system, versatility was evaluated using two different scenarios: changes in payload and changes in group formation. Below, we present the results of the experiments conducted to test both of these scenarios.

4.2.1 Payload

The first scenario aims to test the ability of the system to manipulate different payloads. In this experiment, the amount of payload for a system with three Δp robots is increased gradually from 100 g to 600 g through a set of experiments. These payloads correspond to up to 4 times the weight of each robot. For each of the experiments, position and orientation of the object and the robots are acquired over time using an OptiTrack motion capture system. x , y positions, and the orientation of the object center of mass over time are displayed in Fig. 7-A. Table 2 presents the settling time and steady-state error of the system for different payloads. As expected, performance degrades with increasing payload, while the decentralized controller exhibits similar performance to the centralized controller (patterns of larger steady-state error and smaller settling times are maintained). These results indicate that the settling time and steady-state error of the system increase with the amount of payload for both controllers. This is due to the cancellation of control forces by friction forces experienced by the object. Utilizing a nonlinear control function, $\phi(e)$, may help eliminate this steady-state error, but this is beyond the scope of this article.

4.2.2 Group Formation

Another possible scenario could happen when the shape or placement of the object limits accessible attachment points and consequently robots can not distribute uniformly around the object.

To explore the performance of the system in such scenarios, three different arbitrary formations of three robots around the object are studied. In the first experiment, agents are placed with 60° increments around the object. In this formation none of the virtual agents overlap with a real agent. The second experiment uses 80° increments between the robots. Thus, locations of robots are closer to the locations of the assumed virtual agents. The last experiment studies the response of the system for uniformly distributed agents (with 120° increments) around the object. The results of these experiments are illustrated in Fig. 7-B. As shown in this figure,

Table 2. Settling time and steady-state error for different payloads M

| M (Kg) | Settling time | | | |
|----------|---------------|-------------|-------------|-------------|
| | $e = 15\%$ | | $e = 5\%$ | |
| | Dec. | Cent. | Dec. | Cent. |
| 0.1 | 4.27±0.32 s | 4.52±0.46 s | 5.64±0.49 s | 7.06±0.55 s |
| 0.2 | 5.17±0.51 s | 7.03±0.53 s | 5.85±0.63 s | 7.42±0.79 s |
| 0.3 | 5.88±0.56 s | 7.60±0.72 s | 6.92±0.65 s | 8.69±1.02 s |
| 0.4 | 7.01±0.86 s | >10 s | >10 s | >10 s |
| 0.5 | >10 s | >10 s | >10 s | >10 s |
| 0.6 | >10 s | >10 s | >10 s | >10 s |

| M (Kg) | Steady-state error | |
|----------|-----------------------------|-----------------------------|
| | Dec. | Cent. |
| 0.1 | 2.2±0.53 mm, 1.97±0.54 ° | 1.6±0.34 mm, 0.15± 0.11 ° |
| 0.2 | 4.5±0.68 mm, 2.18±1.75 ° | 3.7±0.72 mm, 1.73± 0.48 ° |
| 0.3 | 9.6±0.84 mm, 6.06±2.62 ° | 8.4±5.50 mm, 4.26±1.87 ° |
| 0.4 | 64±10.40 mm, 7.31±2.93 ° | 48±0.90 mm, 6.18±2.91 ° |
| 0.5 | 193±14.83 mm, 41.58±20.10 ° | 176±21.77 mm, 37.62±24.58 ° |
| 0.6 | 251±21.96 mm, 46.38±27.04 ° | 237±23.60 mm, 42.6±35.90 ° |

robots successfully move a 100 g object to the desired location and orientation for all of the considered formations. Table 3 presents the settling times and steady-state errors for the studied group formation. As observed from these results, the settling time and steady-state error of the system decreases as the distribution of the robots converges to a uniform configuration. This is due to the fact that, for uniform configurations the assumed locations of the virtual agents represents the effect of the real robots more effectively. A similar situation is observed as the group population increases and the probability of coincidence between real and virtual agents becomes higher. This experimental observation verifies our simulation results obtained in [5]. On the other hand, as the attachment points of the agents to the object get closer to each other, the proposed system exhibits a shorter settling time with a larger steady-state error. This is due to an increase in the norm and condition number of the T matrix as described in Section 2.2.

4.3 Robustness

Robustness of the system is characterized by several factors that include [28, 29]: redundancy, decentralized coordination, and structural simplicity of the agents. Redundancy is defined as the capability of the multi-robot system to accomplish the assigned task despite individual failures. Decentralized coordination corresponds to a system property, in which a partial failure will not prevent the system from achieving the goal. This property is not maintained in systems that rely on a leader for coordination since failing the leader (or failing to assign a replacement) will result in failure of the whole group. Simplicity of the agents is another factor that affects the robustness of the system.

The experiments presented in this section are designed

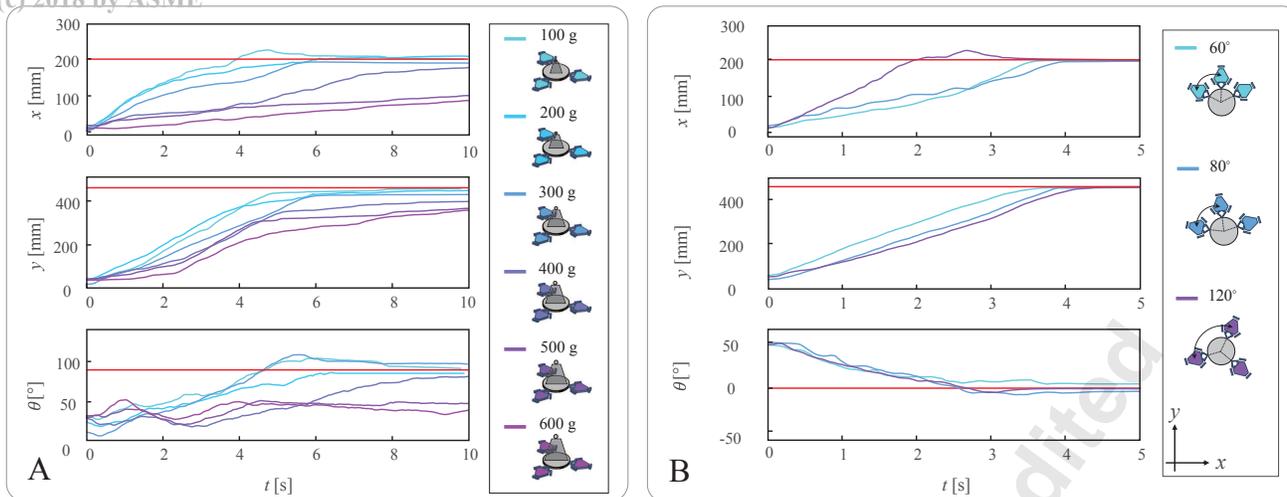


Fig. 7. The x , y positions and the orientation of the object center of mass over time for (A) different payloads ranging from 100 g (cyan) to 600 g (magenta) and (B) different group formations (indicated by different colors). The red line represents the desired object pose

Table 3. Settling time and steady-state error for different group formations

| Increments | Settling time | | | |
|------------|---------------|-------------|-------------|-------------|
| | $e = 15\%$ | | $e = 5\%$ | |
| | Dec. | Cent. | Dec. | Cent. |
| 60° | 3.21±0.20 s | 3.31±0.31 s | 3.33±0.62 s | 3.58±0.73 s |
| 80° | 3.38±0.28 s | 3.53±0.34 s | 3.75±0.48 s | 3.97±0.50 s |
| 120° | 3.52±0.25 s | 3.63±0.35 s | 4.13±0.43 s | 4.29±0.47 s |

| Increments | Steady-state error | |
|------------|--------------------------|--------------------------|
| | Dec. | Cent. |
| 60° | 8.35±1.65 mm, 7.95±3.73° | 4.97±1.61 mm, 6.62±3.43° |
| 80° | 3.93±0.95 mm, 6.48±2.80° | 2.82±0.86 mm, 5.06±3.14° |
| 120° | 1.65±1.05 mm, 5.54±1.53° | 1.33±1.03 mm, 4.79±1.86° |

to investigate the system robustness through redundancy. In this regard, the agents are intentionally programmed to fail at a certain point during each experiment. Figure 8 illustrates the results of failing 3 different sets of 2 robots in a group consisting of 5 agents in total. In each trial, 2 different robots are programmed to not function at time equal to 1 second (a number arbitrarily picked without loss of generality to ensure that failures occur during runtime and before the completion of the task). As depicted in this figure, the system can successfully continue towards minimization of the position error, but the resulting steady-state errors are comparably larger due to an increase in frictional forces. Because of the arm mechanism that robots use to manipulate the object, inactive (failed) agents remain attached to the object through the manipulation phase. This introduces an added weight and friction to the manipulated object. As long as the total payload (including the object and the failed robots)

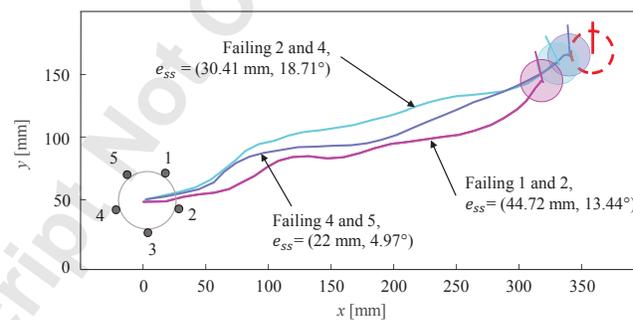


Fig. 8. System configuration for robustness test. Initial robot positions around the object are represented using small gray circles. The desired pose of the object is depicted using dashed red line. In each trial, different selection of the robots involved in the manipulation task are set to be inactive after 1 second, simulating partial power failure. The steady state pose error values for each trial implies the ability of the team to accomplish the task despite failure of 40% of the team

is less than the maximum payload capacity for the remainder of the team, robots can successfully manipulate the object to the goal position. A similar behavior is observed in Fig. 7-A where the response of the system for varying payload is studied. Moreover, since $\phi(e)$ is a PD controller, the steady-state error, e_{ss} , is implicitly defined by $TK_p(e_{ss}) = F_{friction}$. As discussed in [5], in the absence of external disturbances, the object should follow a straight path connecting the initial and final positions. However, the observed variations from the straight path, illustrated in Fig. 8, are due to the friction forces introduced by the failed agents.

4.4 Efficiency

In this article, the term “efficiency” of a system is used to define a metric, which measures the ability of the system

to make the best use of the provided resources for generating the desired output. Although this definition does not precisely match with the classical definition of efficiency in a mechanical system, it allows us to quantitatively capture the advantages and disadvantages of the proposed algorithm.

In addition to manipulation time which was discussed for all of the experiments so far, and is a metric of time efficiency, other metrics could be used to evaluate the energy efficiency of the proposed collective manipulation algorithm. These measures are: 1) The smoothness of the velocity profile of the object, which is used to provide information about the force coordination between agents. In a system with similar frictional effects and coefficients, the one that has a smoother velocity profile is the one that has better coordination between its agents; and 2) The path efficiency of the algorithm, which is defined as the variation from the optimal path between the start and goal points. Since the experiments are conducted in an obstacle-free environment (to hold the focus on the performance of the controller), the optimal path is the straight line that connects the start and goal points (Euclidean shortest path).

4.4.1 Object velocity profile

The velocity profiles of the object during manipulation have been used in [2] as a method to determine the coordination level of the system. As suggested in [2], the difference between the prey (or the object being manipulated) speed and the maximum achievable speed during manipulation can be used to define the coordination level between the agents. The time period in which the prey is carried at the maximum velocity is defined as the period of maintaining coordination.

Since both centralized and decentralized controllers are distributing a PD control law among the agents, the output of $\phi(e)$ is directly related to the distance from the goal position. In a physical system, the output of $\phi(e)$ will be saturated by power limitations of the agents. Thus, the response of the system is expected to converge into three phases of: 1) maximum positive acceleration, 2) an approach phase with a constant velocity, and 3) a final convergence to the desired point. This response type is observable in all the time responses presented in this article.

The velocity profiles suggested by the experimental time responses demonstrate an approximately trapezoidal shape, which is close to the optimal velocity profile for systems with acceleration limits [30, 31]. Although the response of the system in the last phase is affected by frictional forces and controller gains, the approach phase (the constant velocity region in the middle of the trapezoidal velocity curve enables the utilization of a velocity smoothness measure to determine team coordination level. As an example, Fig. 9 presents the speed profile of a 230 g object as it is manipulated by the decentralized controller with a group consisting of 5 agents. In this figure, the shaded region shows the approach phase. In other words, a smoother velocity profile in the approach phase demonstrates a higher coordination level and a continuous cooperative manipulation. In the experiments discussed in this section, deviations from the mean velocity of the ob-

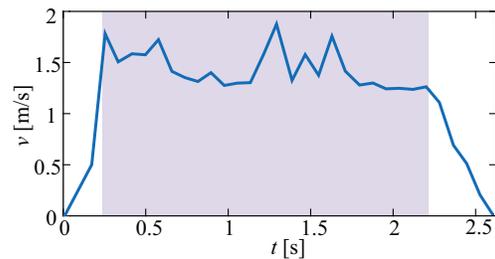


Fig. 9. Object velocity profile during the manipulation

Table 4. Root mean squared deviation of velocity profile for different group population N and different formations

| RMSD | | |
|------------|-----------|-----------|
| N | Dec. | Cent. |
| 2 | 0.74±0.08 | 0.53±0.06 |
| 3 | 0.59±0.06 | 0.49±0.04 |
| 4 | 0.20±0.03 | 0.16±0.05 |
| 5 | 0.17±0.02 | 0.15±0.02 |
| Increments | Dec. | Cent. |
| 60° | 0.56±0.06 | 0.51±0.06 |
| 80° | 0.38±0.06 | 0.35±0.04 |
| 120° | 0.27±0.03 | 0.25±0.05 |

ject, which is a sign of uncoordinated forces in the group, can be demonstrated by variations in the velocity profiles during the approach phase. Such deviations can be demonstrated by utilizing root-mean-square deviation (RMSD), defined as:

$$\text{RMSD}(v) = \sqrt{\frac{\sum_{i=1}^n (\hat{v} - v_i)^2}{n}}, \quad (18)$$

where \hat{v} is the arithmetic mean of the speed signal v , and n is the number of data points. This measure is extensively used in the literature to determine deviations in time series data [32–35].

Table 4 presents the RMSD for the velocity profiles of the object in different experiments studying the effect of group size and formation around the object. Results of this measurement suggest that by increasing the group population or uniformly distributing the agents around the object, smoother velocity profiles can be achieved, which indicates higher coordination levels in the system.

4.4.2 Path efficiency

Path efficiency is defined as the ratio of the shortest path between start and goal points to the distance traveled by the object [36]. Similar to the analogy used for the velocity profiles, path efficiency is a performance index, which shows the

Table 5. Path efficiency for different group population N and different formations

| Path Efficiency | | |
|-----------------|-----------------|-----------------|
| N | Dec. | Cent. |
| 2 | 0.71 ± 0.05 | 0.80 ± 0.04 |
| 3 | 0.82 ± 0.04 | 0.86 ± 0.03 |
| 4 | 0.87 ± 0.03 | 0.90 ± 0.03 |
| 5 | 0.91 ± 0.04 | 0.92 ± 0.02 |
| Increments | | |
| | Dec. | Cent. |
| 60° | 0.78 ± 0.07 | 0.87 ± 0.08 |
| 80° | 0.83 ± 0.05 | 0.90 ± 0.04 |
| 120° | 0.91 ± 0.02 | 0.92 ± 0.03 |

amount of deviation from the shortest path, defined by

$$\eta_{path}(s) = \frac{\|\vec{x}_n - \vec{x}_0\|}{\sum_{i=0}^{n-1} \|\vec{x}_{i+1} - \vec{x}_i\|}, \quad (19)$$

where $s = \{\vec{x}_i : i \in [0, n]\}$ is the set of n data points that are collected through the experiment. Each data point $\vec{x}_i = x\hat{i} + y\hat{j}$ is a point in x - y plane that corresponds to the location of the object at the instant i .

Table 5 presents the path efficiency values for different group populations and formations, respectively. The corresponding values are calculated by considering the shortest path between start and goal positions to be the straight line that connects the two points. As observed from Table 5, increasing the number of agents will result in higher path efficiency, which means less deviation from the shortest path between the start and goal points. Also as discussed before, as the population increases, the decentralized controller response converges to the centralized controller.

Additionally, in a system of three robots and for three different formations around the object, Table 5 suggests that: as the agents become more uniformly distributed, the path efficiency of the system becomes higher. Also, the difference between centralized and decentralized controllers responses becomes less obvious as the agents spread more uniformly around the object. Ultimately, for the uniform distribution of 120° increments, the behavior of the decentralized controller converges to the centralized controller.

4.5 Trajectory tracking

The experiments presented in this manuscript focus on evaluating the performance of the proposed algorithm in an obstacle free environment. Clearly, most practical applications require trajectory tracking as well as simple manipulation. To demonstrate the effectiveness of the method in trajectory control of the object, an experiment is conducted with five robots in which they carry a 230 g object along a sinusoidal path. The results of this experiment are presented

in Figure 10. Although this experiment shows the effectiveness of the algorithm in trajectory tracking, further extensive investigations are required to evaluate its performance.

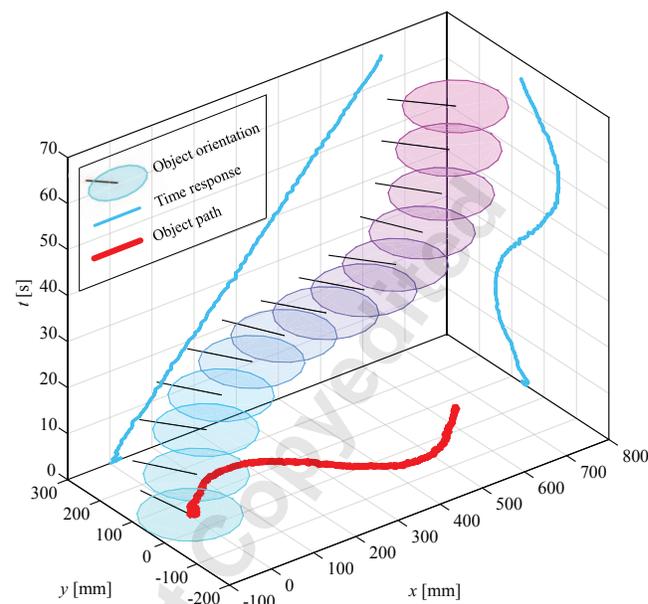


Fig. 10. Timelapse figure of position and orientation of the object being moved along a sinusoidal trajectory. Path of the object center of mass is projected on x - y plane in red. x and y time responses are plotted in x - t and y - t planes respectively. The object color gradient, varying from cyan to magenta, illustrates the passage of time. Orientation of the object at each snapshot is depicted with a black line parallel to its x axis.

5 Conclusions

This article focused on the derivation and experimental analysis of a new decentralized algorithm for cooperative multi-robot object manipulation based on an agent-level force control approach. The presented algorithm utilizes a local Jacobian, which is defined based on the relative position of an agent with respect to the object center of mass (using a corresponding virtual agent), to distribute the output of a control function among the agents. Thus, the architecture of the decentralized controller does not require any information about the population and formation of the group. Some of the advantages of the proposed decentralized approach are: 1) In addition to simultaneous position and orientation control of the object through the course of manipulation, the algorithm also provides the means to implement impedance or force control to allow interaction of the object with the environment [37, 38]. 2) Since construction of the local Jacobians does not require any information about the population and formation of the group, it eliminates the need for any inter-agent communication network. This feature is especially beneficial in real-life implementations with

large number of robots. 3) Coordination between the agents is achieved without relying on a group leader which increases system robustness, and reduces the need for additional algorithmic patches. Consequently, the method could be used as a backbone for multi-behavioral solutions such as decentralized object avoidance in parallel to manipulation [39].

An extensive set of experiments is conducted to evaluate the scalability, versatility, and robustness of the proposed decentralized algorithm. For this purpose, a new robotic platform, $\Delta\rho$, is designed and fabricated. $\Delta\rho$ utilizes a holonomic locomotion system, which provides enough degrees of freedom to exert forces in any planar direction to the object. The experimental setup consists of up to five $\Delta\rho$ robots, which carry payloads up to 600 g (4 times the mass of a single agent). Efficiency of the algorithm is also evaluated by defining quantitative metrics of manipulation time, path efficiency, and deviations in velocity profile. The experiments conducted with different populations and formations of the group, payload values and agent failures proved the scalability, versatility, and robustness of the proposed algorithm. As expected, it is observed that the response of the decentralized controller approaches the response of the centralized controller as the agent locations get closer to the locations of the virtual agents. This could be achieved by increasing the number of agents in the system or uniformly distributing the robots around the object. In general, the decentralized approach demonstrates a shorter settling time due to the fact that the eigenvalues of the transformation matrix T are directly related to the number of agents involved in the manipulation task. Moreover, it is observed that an increase in group population results in shorter manipulation times (higher time-efficiency), smaller steady-state errors, and reduced deviation in path and velocity profiles (indicating higher coordination level and path efficiency), which are predicted by the theoretical model. Experiments with different payloads show that an increase in payload causes larger steady-state errors, which is associated with the balance between static friction forces and the gains of the PD controller. An increase in payload also results in a corresponding increase in manipulation time, which is caused by a reduction in acceleration due to an increase in kinetic friction forces. It is also observed that when the robots are located close to each other, the system shows a faster response and a larger steady-state error, which is due to an increase in the norm and condition number of the transformation matrix T (as explained in Section 2.2). In this study, the experiments are focused on evaluating the performance of the decentralized controller and the experimental setup is designed to accurately represent the behavior of the swarm system after finding and attaching to the object. Throughout the experiments, robots use a non-prehensile fixed arm as their end-effector to apply required planar forces to the object. Adding force sensing arms to measure the applied forces to the object and updating local Jacobians based on force feedback, demonstrating impedance control capabilities of the algorithm, and extending the method to 3-D space are some of the future directions in this research.

Acknowledgements

The authors are grateful to Professor Pratap M. Rao for his thoughtful comments on this manuscript.

References

- [1] Franks, N. R., 1986. "Teams in social insects: group retrieval of prey by army ants (*eciton burchelli*, hymenoptera: Formicidae)". *Behavioral Ecology and Sociobiology*, **18**(6), pp. 425–429.
- [2] Berman, S., Lindsey, Q., Sakar, M. S., Kumar, V., and Pratt, S. C., 2011. "Experimental study and modeling of group retrieval in ants as an approach to collective transport in swarm robotic systems". *Proceedings of the IEEE*, **99**(9), pp. 1470–1481.
- [3] Krieger, M. J., Billeter, J.-B., and Keller, L., 2000. "Ant-like task allocation and recruitment in cooperative robots". *Nature*, **406**(6799), pp. 992–995.
- [4] Camazine, S., 2003. *Self-organization in biological systems*. Princeton University Press.
- [5] Faal, S. G., Kalat, S. T., and Onal, C. D., 2016. "Towards collective manipulation without inter-agent communication". In Proceedings of the 31st Annual ACM Symposium on Applied Computing, ACM, pp. 275–280.
- [6] Kube, C. R., and Bonabeau, E., 2000. "Cooperative transport by ants and robots". *Robotics and autonomous systems*, **30**(1), pp. 85–101.
- [7] Wan, W., Fukui, R., Shimosaka, M., Sato, T., and Kuniyoshi, Y., 2012. "Cooperative manipulation with least number of robots via robust caging". In Advanced Intelligent Mechatronics (AIM), 2012 IEEE/ASME International Conference on, IEEE, pp. 896–903.
- [8] Kobayashi, Y., and Hosoe, S., 2012. "Cooperative enclosing and grasping of an object by decentralized mobile robots using local observation". *International Journal of Social Robotics*, **4**(1), pp. 19–32.
- [9] Campo, A., Nouyan, S., Birattari, M., Groß, R., and Dorigo, M., 2006. "Negotiation of goal direction for cooperative transport". In *Ant Colony Optimization and Swarm Intelligence*. Springer, pp. 191–202.
- [10] Rubenstein, M., Cabrera, A., Werfel, J., Habibi, G., McLurkin, J., and Nagpal, R., 2013. "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, International Foundation for Autonomous Agents and Multiagent Systems, pp. 47–54.
- [11] Groß, R., and Dorigo, M., 2004. "Group transport of an object to a target that only some group members may sense". In *Parallel Problem Solving from Nature-PPSN VIII*, Springer, pp. 852–861.
- [12] Wang, Z., and Schwager, M., 2014. "Multi-robot manipulation without communication". In *International Symposium on Distributed Autonomous Robotic Systems (DARS)*.
- [13] Wang, Z., and Schwager, M., 2016. "Kinematic multi-robot manipulation with no communication us-

- ing force feedback”. In IEEE International Conference on Robotics and Automation (ICRA).
- [14] Kube, C. R., and Zhang, H., 1993. “Collective robotics: From social insects to robots”. *Adaptive behavior*, **2**(2), pp. 189–218.
- [15] Zavlanos, M. M., Jadbabaie, A., and Pappas, G. J., 2007. “Flocking while preserving network connectivity”. In Decision and Control, 2007 46th IEEE Conference on, IEEE, pp. 2919–2924.
- [16] Becker, A., Habibi, G., Werfel, J., Rubenstein, M., and McLurkin, J., 2013. “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, IEEE, pp. 520–527.
- [17] Craig, J. J., 2005. *Introduction to robotics: mechanics and control*, Vol. 3. Pearson Prentice Hall Upper Saddle River.
- [18] Hogan, N., 1984. “Impedance control: An approach to manipulation”. In American Control Conference, 1984, IEEE, pp. 304–313.
- [19] Li, Z., Hsu, P., and Sastry, S., 1989. “Grasping and coordinated manipulation by a multifingered robot hand”. *The International Journal of Robotics Research*, **8**(4), pp. 33–50.
- [20] Golan, J. S., 2012. “Moore–penrose pseudoinverses”. In *The Linear Algebra a Beginning Graduate Student Ought to Know*. Springer, pp. 441–452.
- [21] Rubenstein, M., Ahler, C., and Nagpal, R., 2012. “Kilobot: A low cost scalable robot system for collective behaviors”. In Robotics and Automation (ICRA), 2012 IEEE International Conference on, IEEE, pp. 3293–3298.
- [22] Bonani, M., Longchamp, V., Magnenat, S., Rétonnaz, P., Burnier, D., Roulet, G., Vaussard, F., Bleuler, H., and Mondada, F., 2010. “The marxbot, a miniature mobile robot opening new perspectives for the collective-robotic research”. In Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, IEEE, pp. 4187–4193.
- [23] Birkmeyer, P., Peterson, K., and Fearing, R. S., 2009. “Dash: A dynamic 16g hexapedal robot”. In Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, IEEE, pp. 2683–2689.
- [24] Kalat, S. T., Faal, S. G., Celik, U., and Onal, C. D., 2015. “Tribot: A minimally-actuated accessible holonomic hexapedal locomotion platform”. In Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, IEEE, pp. 6292–6297.
- [25] Faal, S. G., Chen, F., Tao, W., Agheli, M., Tasdighikalat, S., and Onal, C. D., 2016. “Hierarchical kinematic design of foldable hexapedal locomotion platforms”. *Journal of Mechanisms and Robotics*, **8**(1), p. 011005.
- [26] Wang, Z., Yang, G., Su, X., and Schwager, M., 2016. “Ouijabots: Omnidirectional robots for cooperative object transport with rotation control using no communication”. In Proc. Int. Conf. Distrib. Auton. Robot. Syst.
- [27] Virtual coordination in collective object manipulation. <http://softrobotics.wpi.edu/tmp/WPI-SRL-2016-ColMan01.mp4>.
- [28] Şahin, E., 2004. “Swarm robotics: From sources of inspiration to domains of application”. In *Swarm robotics*. Springer, pp. 10–20.
- [29] Olariu, S., and Zomaya, A. Y., 2005. *Handbook of bioinspired algorithms and applications*. CRC Press.
- [30] Velenis, E., and Tsiotras, P., 2005. “Optimal velocity profile generation for given acceleration limits: theoretical analysis”. *system*, **2**, p. 5.
- [31] Haddad, M., Khalil, W., and Lehtihet, H., 2010. “Trajectory planning of unicycle mobile robots with a trapezoidal-velocity constraint”. *Robotics, IEEE Transactions on*, **26**(5), pp. 954–962.
- [32] NETO, P. d. M., Araújo, R. d. A., Petry, G. G., Ferreira, T. A., and Vasconcelos, G. C., 2007. “Hybrid swarm system for time series forecasting”. *VI Encontro Nacional de Inteligência Artificial (ENIA)*.
- [33] Armstrong, J. S., and Collopy, F., 1992. “Error measures for generalizing about forecasting methods: Empirical comparisons”. *International journal of forecasting*, **8**(1), pp. 69–80.
- [34] Willmott, C. J., 1981. “On the validation of models”. *Physical geography*, **2**(2), pp. 184–194.
- [35] Hyndman, R. J., and Koehler, A. B., 2006. “Another look at measures of forecast accuracy”. *International journal of forecasting*, **22**(4), pp. 679–688.
- [36] Habibi, G., Schmidt, L., Jellins, M., and McLurkin, J., 2016. “K-redundant trees for safe and efficient multi-robot recovery in complex environments”. In *Robotics Research*. Springer, pp. 149–165.
- [37] Kalat, S. T., Faal, S. G., and Onal, C. D., 2017. “Scalable collective impedance control of an object via a decentralized force control method”. In American Control Conference (ACC), 2017, IEEE, pp. 2680–2686.
- [38] Kalat, S. T., 2017. “Virtual coordination in collective object manipulation”. Master’s thesis, Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA, 01609.
- [39] Faal, S. G., Kalat, S. T., and Onal, C. D., 2017. “Decentralized obstacle avoidance in collective object manipulation”. In Adaptive Hardware and Systems (AHS), 2017 NASA/ESA Conference on, IEEE, pp. 133–138.