2017-04-26

Virtual Coordination in Collective Object Manipulation

Shadi Tasdighi Kalat
Worcester Polytechnic Institute

Virtual Coordination in Collective Object Manipulation

by

Shadi T. Kalat

A Thesis

Submitted to the faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

in

Mechanical Engineering

May 2017

APPROVED:

_____

Professor Cagdas. D. Onal, Thesis Advisor

_____

Professor Gregory S. Fischer, Graduate Committee Member

_____

Professor Umberto Mosco, Graduate Committee Member

_____

Professor Carlo Pinciroli, Graduate Committee Member

_____

Professor Marcus Sarkis-Martins, Graduate Committee Member

_____

Professor Pratrap M. Rao, ME Graduate Committee Representative

Abstract

Inspired by nature, swarm robotics aims to increase system robustness while utilizing simple agents. In this work, we present a novel approach to achieve decentralized coordination of forces during collective manipulation tasks resulting in a highly scalable, versatile, and robust solution. In this approach, each robot involved in the collective object manipulation task relies on the behavior of a cooperative "virtual teammate" in a fully decentralized architecture, regardless of the size and configuration of the real team. By regulating their actions with their corresponding virtual counterparts, robots achieve continuous pose control of the manipulated object, while eliminating the need for inter-agent communication or a leader-follower architecture. To experimentally study the scalability, versatility, and robustness of the proposed collective object manipulation algorithm, a new swarm agent, $\Delta\rho$ (Delta-Rho) is introduced which is able to apply linear forces in any planar direction. Efficiency and effectiveness of the proposed decentralized algorithm are investigated by quantitative performance metrics of settling time, steady-state error, path efficiency, and object velocity profiles in comparison with a force-optimal centralized version that requires complete information.

Employing impedance control during manipulation of an object provides a mean to control its dynamic interactions with the environment. The proposed decentralized algorithm is extended to achieve a desired multi-dimensional impedance behavior of the object during a collective manipulation without inter-agent communication. The proposed algorithm extension is built upon the concept of "virtual coordination" which demands every agent to locally coordinate with one virtual teammate. Since the real population of the team is unknown to the agents, the

resultant force applied to the manipulated object would be directly scaled with the team population. Although this scaling effect proves useful during position control of the object, it leads to a deviation from the desired dynamic response when employed in an impedance control scheme. To minimize such deviations, a gradient descent algorithm is implemented to determine a scaling parameter defined on the control action. The simulation results of a multi-robot system with different populations and formations verify the effectiveness of the proposed method in both generating the desired impedance response and estimating the population of the group. Eventually, as two case studies, the introduced algorithm is used in robotic collective manipulation and human-assistance scenarios.

Simulation and experimental results indicate that the proposed decentralized communication-free algorithm successfully performs collective manipulation in all tested scenarios, and matches the performance of the centralized controller for increasing number of agents, demonstrating its utility in communication-limited systems, remote environments, and access-limited objects.

Acknowledgements

First and foremost, I would like to express my gratitude to my advisor Professor Cagdas D. Onal, director of the WPI Soft Robotics Laboratory, for continuosly supporting and advising me during my research. Additionally, I would like to thank my thesis committee members for generously offering their time, support and guidance that helped me to improve this manuscript.

I am specially thankful to Siamak G. Faal, for all his thechnical and moral support. This work would have not been possible without his help.

I would also like to thank Mechanical Engineering department and Academic & Research Computing group for supporting my studies at Worcester Polytechnic Institute.

Last but not the least, I would always be grateful to my family for their unconditional love and support throughout every decision I made.

Contents

List of Figures

List of Tables

Nomenclature

γ	Step size in Gradient Descent method
$\phi(e)$	Control function
τ_k	Traction force on the k^{th} wheel of the robot
CoM	Center of Mass
e	Object error defined as the difference between current and desired pose
e_a	Agent attitude
F	Overall force vector required for manipulating the object
F^*	Required force vector computed from the desired impedance behavior of the object
f_i	The force vector applied by the i^{th} agent
F_{ext}	External forces applied to the object from the environment
I	Mass moment of inertia of the manipulated object
J	Full Jacobian of the system
J_A	Agent's Jacobian used in robot control

J_i Local Jacobian assumed by the i^{th} agent

m Mass of the manipulated object

N Number of agents in the system

r_i Attachment point of the i^{th} agent measured from CoM of the object

$RMSD$ Root Mean Square Deviation

s_i Scaling facor of force for i^{th} agent

T Tranfromation matrix mapping the output of $\phi(e)$ to the force vector that is applied to the CoM of the object

x Object pose

x_0 Object initial pose

x_d Object desired pose

Chapter 1

Introduction

1.1 Collective Manipulation

Swarm systems are capable of demonstrating global intelligent behavior that emerges from simple rules followed by a large number of agents with limited abilities and intelligence. One of the interesting and 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 observation inspired researchers to study the underlying rules that may govern synergistic tasks of such collective behavior [4, 17], to design flexible, robust and effective robotic systems [29].

In robotics, unlike classical frameworks which focus on increasing dexterity and intelligence of a single well-instrumented robot, swarm systems allow utilization of many low-cost, small, and simple robots to realize complex tasks. Due to their structural simplicity, population size and distributed nature, swarm systems are more scalable, flexible, and robust in comparison to single-agent solutions [9]. On the other hand, utilization of a swarm framework introduces a suite of challenges

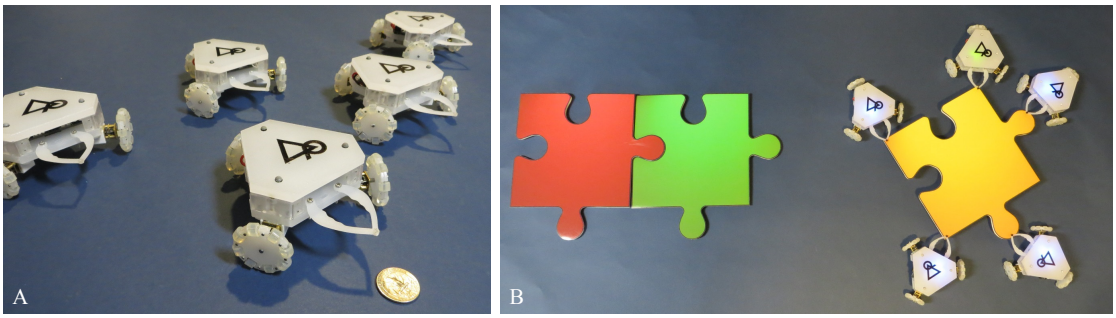

Fig. 1.1. (A) Fabricated $\Delta\rho$ swarm agents. (B) A multi robot system consisting five $\Delta\rho$ robots as they are manipulating a puzzle piece

on integrating communication, computation, and control, especially for increasing number of agents.

This research introduces a decentralized force control algorithm and analyzes its application in collective object manipulation with a custom multi-robot system. 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. Consequently, force coordination between each agent and its respective virtual teammate originates the global 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 multi-robot manipulation 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 for and controlling positions of agents) generates a degree

of mechanical compliance on 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 in 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 designed to be fabricated by laser machining of interlocking 2-D profiles of uniform thickness, which significantly reduces the fabrication cost and time. Fig. 1.1-A illustrates the $\Delta\rho$ platforms and Fig. 1.1-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, 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 highly scalable communication-free collective tasks with broad application areas such as search-and-rescue, construction, and warehouse automation.

1.1.1 Related work

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. In general, collective manipulation methods in robotic systems can be categorized into two main groups: 1) Manipulating the object with push/pull forces that are directed towards the goal position (force closure), and 2) Caging approaches, in which the object is confined by the agents as they move towards the goal (form closure).

Pushing/pulling the object towards the goal position is one of the earliest solutions developed for the realization of collective manipulation. As one of the early attempts, [31] developed a rather complicated subsumption network of behaviors for collective transportation of an object using simple robots. Each robot is governed by two fundamental behaviors: 1) "Avoid interfering with another robot", and 2) "Work toward a common task while observing the first rule". A "follow" behavior is introduced for a robot to follow another robot and eventually help the group in exerting pushing forces to the object. This algorithm is experimentally validated with five box-pushing robots. Each robot comprised a differential drive locomotion system and was equipped with a pair of (left and right) infrared sensors to detect obstacles and a pair of left and right photocells to detect the brightly lit box as the object.

The performance of the approach is affected by the population of the group, the location of the center of mass and the shape of the object. Although this simple approach seems effective in finding the optimal path (a straight line between the initial and goal positions of the object in the absence of obstacles), in the presented form, it does not provide simultaneous control of the position and orientation of the object during transport. Thus, after finishing the position control phase, the robots should move tangent to the object to correct the orientation. This additional step increases the manipulation time. Simultaneous control of position and orientation

is achievable by introducing a pre-planning phase to the algorithm at a cost of increased computation time. An interesting decentralized cooperative transport algorithm is introduced in [11], where the robots only push the object at positions where the direct line of sight to the goal is occluded by the object. Although this approach does not require any communication between agents, it also fails to control the orientation of the object.

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, [30] addresses the issue of stagnation and proposes a recovery mechanism, which utilizes the application of random forces by either realigning the force angles or repositioning the pushing force.

In contrast to pushing methods, pulling strategies require secure attachment of the agents to the object. Due to their appropriate physical mechanisms to carry out grasping, biological agents such as ants utilize pulling strategy as the most common way of transportation [40]. To address this demand, a robotic agent, s-bot, was introduced in [13] and has been used in studies, which incorporate pulling of the object during transportation. [19] utilized the s-bot model in a simulation environment to train an artificial neural network to produce solitary transport and group transport behaviors. Their simulation results show that a group of robots that are not necessarily aware of each other, can still demonstrate effective group transport.

Another common method in collective object transport research is the caging approach. In this method, several agents physically confine the object and the transportation task is achieved by transferring the obtained cage to the desired location [36, 37, 44, 45, 48]. As an example of caging, [43] proposed a control law, which regulates the distance between three adjacent agents as they move; thus

allowing the agents to maintain a prescribed formation to trap the object in their midst while moving towards the goal. [49] used the caging strategy and develop a set of decentralized algorithms that lead to simple, first-order, potential field based controllers for multi-robot manipulation. Their algorithm is composed of two steps: 1) Approaching the object which is achieved by utilizing a potential field in which the object is the attractor and other robots are repulsive. 2) After the execution of the first step, agents move along the perimeter of the object looking for conditions that might satisfy the requirements of form closure. A similar approach with three control modes of approach, organization and transportation is also discussed in [42]. In approach mode, the robots move towards the object by following an attractive potential field centered at the object location. After reaching the object, each robot enters into organization mode, where it tries to move away from its neighbors by adding repulsive potentials imposed by its neighbors to the main potential field. Finally, each robot enters into transportation mode after it senses a local object closure and transports the object based on an added transportation potential with a much lower intensity and centered at the goal.

In form closure approaches, [47] 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 [28]. 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 subtasks 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.

Another approach in this category is introduced by [39] where an over-damped dynamics model was utilized to describe the translations and rotations of the object under the effects of the applied push/pull forces with a first-order differential equation. The object was assumed to be constrained to planar motion and the forces exerted by each robot were assumed to be directly related to its velocity minus its maximum achievable speed. Eventually, the velocity of each agent was determined in the direction of minimizing the position error. Based on this model, [39] concluded that with enough agents to overcome the static friction and a bounded initial orientation, simple push/pull approach guarantees successful manipulation.

Regardless of the strategy used for collective manipulation, successful force/motion coordination requires an agreement on the goal position, also known as goal visibility. This agreement can be obtained either by propagating the goal location to all the involved agents or relying on group leaders, which are assumed to know the goal location. The latter approach requires a consensus strategy among agents. An example of such an approach is described in [10, 39], where a consensus based coordination algorithm is proposed, in which not all the robots can detect the goal. Based on this method, agents which are aware of the goal location will move towards it, while other agents try to minimize their direction error with respect to their neighbors. Another consensus approach is studied in [20] 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 align its body towards the goal location and set its speed to the maximum value. Blind robots can perceive traction forces and movement and consequently determine the desired orientation of their body and move in that direction.

Consensus can also be attained via physical feedback from the object. This approach is studied in [50,51], 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 different 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 [30, 31] 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 [53] 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, [3] 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 Contributions

In this work, we took a top-down approach to calculate the individual agent forces from the desired collective behavior of the system. This approach differs from related works in 3 main aspects: 1) Instead of imposing the required motions to the agents to move the object, our proposed decentralized algorithm distributes a desired force and moment between agents without requiring any information about the group. Thus, the algorithm is not only suitable for collective manipulation, but it can also be utilized in force and impedance control of the object, which is a necessary feature in cooperative assembly and construction tasks; 2) While the algorithm provides simultaneous control of position and orientation, force coordination does not require any inter-agent communication or group leaders; and 3) Our force-based approach in controlling the object pose is versatile for utilization in homogeneous as well as heterogeneous groups.

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. We assume that the robots do not know the shape or mass of the object, or the group population or configuration. In addition, we assume that each agent can monitor its own pose and the pose of the object center of mass (CoM). Experimentally, we use a motion capture system to relay this information to each agent. We believe these are reasonable assumptions to focus solely on the performance of the collective manipulation algorithm itself. For real-world implementation, an ad-

ditional coordination layer could be studied to achieve distributed perception relying only on local information, but this is beyond the scope of this manuscript.

Although cooperative manipulation is achievable through basic algorithms, complete control on the trajectory of the manipulated object and its interactions with the environment requires more advanced approaches.

This manuscript also presents a decentralized algorithm to achieve a desired multi-dimensional impedance behavior of the object during a collective manipulation without inter-agent communication. The proposed algorithm introduces the concept of "virtual coordination" arising from an agent-level assumption, which demands every agent to locally coordinate with one "virtual teammate". Since the real population of the team is unknown to the agents, the resultant force applied to the manipulated object would be directly scaled with the team population. Although this scaling effect proves useful during position control of the object, it leads to a deviation from the desired dynamic response when employed in an impedance control scheme. To minimize such deviations, a gradient descent algorithm is implemented to determine a scaling parameter defined on the control action. The simulation results of a multi-robot system with different populations and formations verify the effectiveness of the proposed method in both generating the desired impedance response and estimating the population of the group. Eventually, as two case studies, the introduced algorithm is used in robotic collective manipulation and human-assistance scenarios.

Chapter 2

Methodology

2.1 Theory

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 manuscript are adopted form [12]. Based on these notations, a transformation A from coordinate frame $\{i\}$ to coordinate frame $\{j\}$ is denoted by ${}^{j}_{i}A$. Similarly, ${}^{i}v$ illustrates vector v defined in coordinate frame $\{i\}$. It is assumed that the coordinate frame $\{O\}$ is attached to the CoM of the object and the vector ${}^{O}r_i$ defines the attachment point of agent i measured from CoM of the object in $\{O\}$. Assuming that the robots are only able to apply forces but not moments, vector ${}^{O}f_i$ defines the force applied by the i^{th} agent in coordinate frame $\{O\}$. The error vector, ${}^{O}e$, is composed of linear and angular differences between desired and current object positions in $\{O\}$ and defined as:

$$ {}^{O}e = (x_d - x_o){}^{O}\hat{i} + (y_d - y_o){}^{O}\hat{j} + (\theta_d - \theta_o){}^{O}\hat{k} \tag{2.1} $$

Equation (2.2) describes the differential equations of motion for an object with mass m and mass moment of inertia I in the body-fixed (non-inertial) reference frame $\{O\}$. The vectors ${}^{O}v_o$ and ${}^{O}f_i$ represent the velocity of the object and the forces applied by each agent, respectively.

$$\begin{bmatrix} m({}^{O}\dot{v}_o + \dot{\theta}_o {}^{O}\hat{k} \times^{O} v_o) \\ I\ddot{\theta}_o {}^{O}\hat{k} \end{bmatrix} = \sum_{i=1}^{N} \begin{bmatrix} {}^{O}f_i \\ {}^{O}r_i \times^{O} f_i \end{bmatrix} \tag{2.2}$$

where N represents the total number of agents that are involved in the task. Unless noted otherwise, throughout the rest of this manuscript, 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.(2.2) can be reformulated as:

$$\sum_{i=1}^{N} \begin{bmatrix} f_i \\ r_i \times f_i \end{bmatrix} = J \begin{bmatrix} f_1^T & f_2^T & \cdots & f_N^T \end{bmatrix}^T = F, \tag{2.3}$$

where the matrix J is the full Jacobian of the system. In this equation r_{xi} and r_{yi} are the x and y components of the r_i vector respectively. Full system Jacobian is written explicitly as:

$$J = \begin{bmatrix} 1 & 0 & 1 & 0 & \cdots & 1 & 0 \\ 0 & 1 & 0 & 1 & \cdots & 0 & 1 \\ -r_{y1} & r_{x1} & -r_{y2} & r_{x2} & \cdots & -r_{yN} & r_{xN} \end{bmatrix} \tag{2.4}$$

2.1.1 Centralized Controller

The objective is to define all the components of f_i such that the norm of the error vector converges to zero in finite time. This can be achieved by equating $F =$

$\begin{bmatrix} F_x & F_y & M_z \end{bmatrix}^T$ to the output of a control function, $\phi(e)$. Although any linear or nonlinear controller can be utilized as the control function, considering system dynamics, $\phi(e)$ 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 (2.3) 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:

$$\begin{bmatrix} f_1^T & \cdots & f_N^T \end{bmatrix}^T = J^+\phi(e) = J^+(K_p e + K_d \dot{e}), \tag{2.5}$$

where J^+ is the Moore-Penrose pseudoinverse of the Jacobian matrix J.

Derivation of the decentralized controller is inspired from a centralized controller that utilizes the full Jacobian of the system and Moore-Penrose pseudoinverse [18] to effectively distribute the control action among the team agents. The details on formulation of the centralized controller are discussed in Section 2.3. Throughout the rest of this manuscript the terms "pseudoinverse" and "Moore-Penrose pseudoinverse" are used interchangeably.

2.2 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 [16] over simulation results. As noted in [16], 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, which results in a direct mapping between the vectors F and f_i.

This direct mapping is achieved if the matrix T is positive definite (as explained in Section 2.3). 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 center of mass (CoM) of the object as shown in Fig. 2.1-A. Thus, each agent can assume the following local Jacobian matrix J_i:

$$J_i = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ -r_{yi} & r_{xi} & r_{yi} & -r_{xi} \end{bmatrix} \tag{2.6}$$

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 [50]. Thus, employing J_i in deriving the local forces eliminates inquiring information about the real team composition. Finally, by following a similar approach to Eqn.(2.8) and substituting the feedback control law $\phi(e)$ with a PD control function, the complete decentralized control law for each agent can be written as:

$$f_i = {J_i}^{+}\phi(e) = {J_i}^{+}(K_p e + K_d \dot{e}), \tag{2.7}$$

where ${J_i}^{+}$ is the Moore-Penrose pseudoinverse of the local Jacobian matrix J_i as shown in Fig. 2.1-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. Derivation of the motor torque inputs based on the calculated force for each agent is described in Section 3.1 .

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 DoF, the extra DoF provided by the holonomic locomotion system of $\Delta\rho$ constitutes its null space, which can be utilized to control the relative angle between the robot and the object as a secondary goal. We use this 1-DoF redundancy to keep the robots at an approximately constant relative orientation (normal to the object perimeter) as described in Section 3.1. The effect of the controller used to correct the orientation of the robots with respect to the object is depicted in Fig. 2.1-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.

2.3 Controller Performance

The objective is to define all the components of f_i such that the norm of the error vector converges to zero in finite time. This can be achieved by equating $F =$

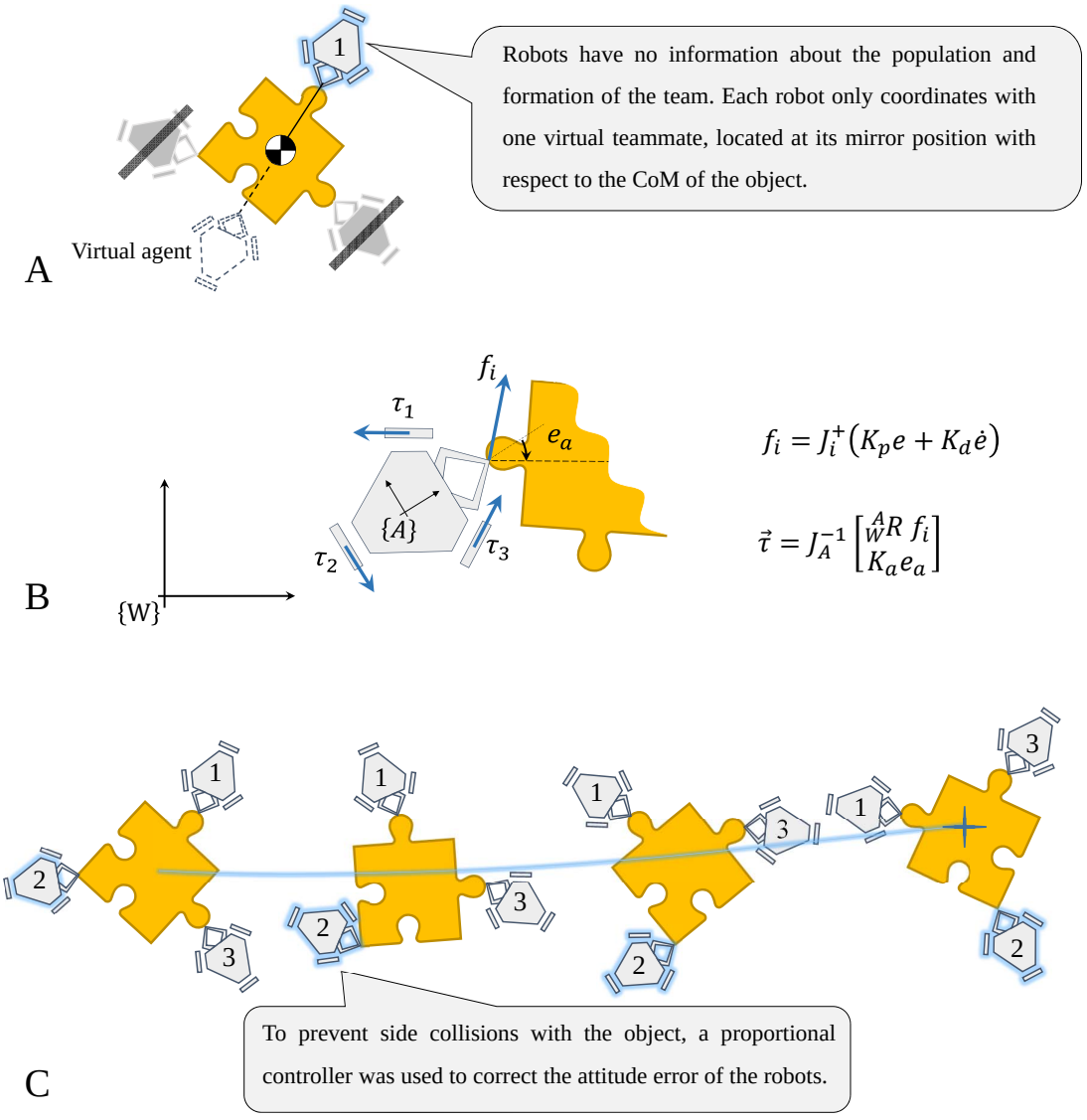

Fig. 2.1. (A) System virtual configuration from each robots view. This virtual configuration for each robot consists of its self and a virtual teammate located at the robots mirror position with respect to the CoM of the object. (B) Utilization of the virtual agents eliminates the need for inter-agent communications and enables each robot to calculate its force vector 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 seen in 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

$\begin{bmatrix} F_x & F_y & M_z \end{bmatrix}^T$ to the output of a control function, $\phi(e)$. Although any linear or nonlinear controller can be utilized as the control function, considering system dynamics, $\phi(e)$ 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.(2.3) 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:

$$\begin{bmatrix} f_1^T & \cdots & f_N^T \end{bmatrix}^T = J^+\phi(e) = J^+(K_p e + K_d \dot{e}), \tag{2.8}$$

where J^+ is the Moore-Penrose pseudoinverse of the Jacobian matrix J.

The resultant applied force to the CoM of the object for each controller can be obtained by substituting the forces obtained using the corresponding control equations into Eqn.(2.3). For the case of the centralized approach this substitution yields:

$$\sum_{i=1}^{N} \begin{bmatrix} f_i \\ r_i \times f_i \end{bmatrix} = J \begin{bmatrix} f_1^T & \cdots & f_N^T \end{bmatrix}^T = J[J^+\phi(e)] \tag{2.9}$$

As observed in Eqn.(4.6), the rows of the Jacobian matrix J are linearly independent. Thus multiplication of J by its pseudo-inverse results in an identity matrix and equation Eqn.(2.9) reduces to:

$$\sum_{i=1}^{N} \begin{bmatrix} f_i \\ r_i \times f_i \end{bmatrix} = \phi(e) \tag{2.10}$$

Thus, 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, it is possible to find the total force applied to the CoM of the object for the decentralized controller. Substituting Eqn. (2.7) into Eqn.(2.3) yields:

$$\sum_{i=1}^{N} \begin{bmatrix} f_i \\ r_i \times f_i \end{bmatrix} = J \begin{bmatrix} f_1^T & \dots & f_N^T \end{bmatrix}^T = J \begin{bmatrix} K_1 \\ K_2 \\ \vdots \\ K_N \end{bmatrix} \phi(e), \tag{2.11}$$

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

$$K_i = \frac{1}{2\Re_i} \begin{bmatrix} k_{i_{11}} & k_{i_{12}} & k_{i_{13}} \\ k_{i_{21}} & k_{i_{22}} & k_{i_{23}} \end{bmatrix}, \tag{2.12}$$

where:

$\Re_i = (a_{xi} - r_{xi})^2 + (a_{yi} - r_{yi})^2$

$k_{i_{11}} = \Re_i + {a_{yi}}^2 - {r_{yi}}^2$

$k_{i_{12}} = -(a_{xi} + r_{xi})(a_{yi} - r_{yi})$

$k_{i_{13}} = 2(a_{yi} - r_{yi})$

$k_{i_{21}} = -(a_{xi} - r_{xi})(a_{yi} + r_{yi})$

$k_{i_{22}} = \Re_i + {a_{xi}}^2 - {r_{xi}}^2$

$k_{i_{23}} = -2(a_{xi} - r_{xi}).$

In the above equation, a_{xi} and a_{yi} are the x and y coordinates of the location of the virtual agent i that are defined in the object coordinate frame $\{O\}$. In general,

$\begin{bmatrix} K_1 & K_2 & \cdots & K_N \end{bmatrix}^T$ will not be equal to the pseudoinverse of J and the right hand side of Eqn. Eqn.(2.11) 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 CoM of the object, is defined as:

$$T\phi(e) = J \begin{bmatrix} K_1 \\ K_2 \\ \vdots \\ K_N \end{bmatrix} \phi(e). \tag{2.13}$$

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 CoM of the object. Although $\vec{a}_i$ vectors can have any arbitrary values as long as they yield to a valid T matrix, the formulation presented in this study assumes that the virtual agent for i^{th} agent is located at its mirror position with respect to the CoM of the object. Substituting values of $\vec{a}_i = -\vec{r}_i$ into Eqn.(2.12) yields:

$$K_i = \frac{1}{2} \begin{bmatrix} 1 & 0 & -r_{yi}/({r_{xi}}^2 + {r_{yi}}^2) \\ 0 & 1 & r_{xi}/({r_{xi}}^2 + {r_{yi}}^2) \end{bmatrix}. \tag{2.14}$$

Finally, substituting K_i values into Eqn.(2.13) yields to the transformation ma-

trix T of proposed decentralized algorithm:

$$T = \frac{1}{2}\begin{bmatrix} N & 0 & -\sum_{i=1}^{N} r_{yi}/({r_{xi}}^2 + {r_{yi}}^2) \\ 0 & N & \sum_{i=1}^{N} r_{xi}/({r_{xi}}^2 + {r_{yi}}^2) \\ -\sum_{i=1}^{N} r_{yi} & \sum_{i=1}^{N} r_{xi} & N \end{bmatrix} \tag{2.15}$$

And the corresponding eigenvalues of the matrix T are:

$$\lambda = \begin{bmatrix} N & N+\delta & N-\delta \end{bmatrix}, \tag{2.16}$$

where δ is defined as:

$$\delta = \sqrt{\sum_{i=1}^{N} \frac{r_{yi}}{{r_{xi}}^2 + {r_{yi}}^2} \sum_{i=1}^{N} r_{yi} + \sum_{i=1}^{N} \frac{r_{xi}}{{r_{xi}}^2 + {r_{yi}}^2} \sum_{i=1}^{N} r_{xi}} \tag{2.17}$$

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 and forces applied to CoM of the object. 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 CoM of the object 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 manuscript, 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.(2.17). 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:

$$\delta = \sqrt{\Big(\sum_{i=1}^{N}\sin(\theta_i)\Big)^2 + \Big(\sum_{i=1}^{N}\cos(\theta_i)\Big)^2} \tag{2.18}$$

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.

Chapter 3

Experimental Validation

3.1 Delta-Rho Robotic Platform

The large population of agents involved in a swarm system demands low-cost platforms to serve as team members. Consequently most of the potential robotic swarm agents introduced so far utilize a simplistic locomotion system. Some of the common approaches for locomotion system designs includes: vibration based locomotion [38], differential drives [6], and two and three degrees-of-freedom (DoF) legged locomotion systems [5, 15, 27]. Although our earlier work [27] 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

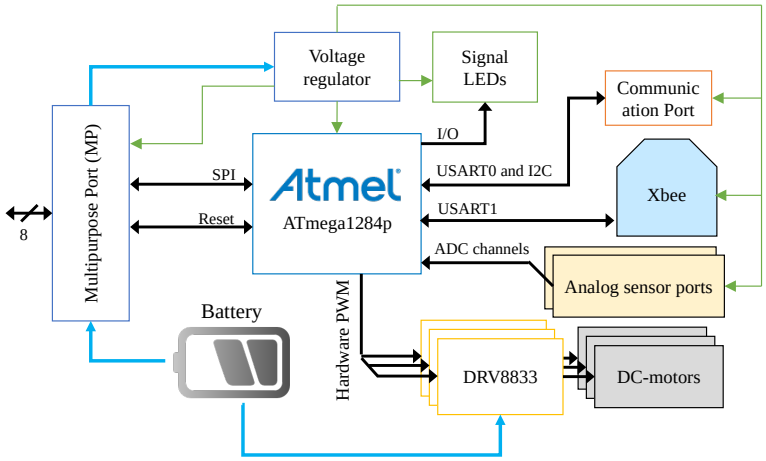

Fig. 3.1. System architecture of $\Delta\rho$. Black, blue and green arrow indicate data, power and regulated voltage lines, respectively

what follows.

3.1.1 Robot Design

$\Delta\rho$ 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. The architecture of the complete experimental system is depicted in Fig. 3.1 and a rendered CAD model with an exploded view of $\Delta\rho$ is illustrated in Fig. 3.2.

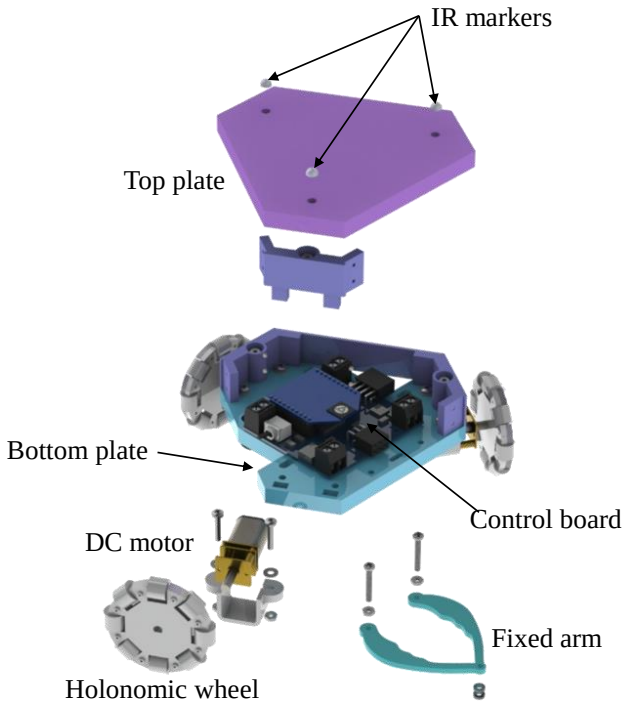

Fig. 3.2. An exploded view of the $\Delta\rho$ robot

3.1.2 Robot Control

Below, we describe the derivation of the motor torque inputs corresponding to each wheel of the holonomic drive system to provide the necessary manipulation forces calculated by the collective manipulation algorithm as well as an additional orientation correction moment around the attachment point of the robot. The dimensional parameters used in this formulation are described in Fig. 3.3.

A geometric mapping between traction forces of the robot wheels, $[\tau_1, \tau_2, \tau_3]^T$, and force and moment vectors at the tip of the end effector, $[T_x, T_y, C_z]^T$ can be described by:

$$\begin{bmatrix} T_x \\ T_y \\ C_z \end{bmatrix} = \sum_{w=1}^{3} \begin{bmatrix} -\sin(\alpha_w) \\ \cos(\alpha_w) \\ \rho_{wx}\cos(\alpha_w) - \rho_{wy}\sin(\alpha_w) \end{bmatrix} \tau_w, \tag{3.1}$$

where $\alpha_w \in \{\pi/3, \pi, -\pi/3\}$ is the angle of the wheel w axis measured form positive

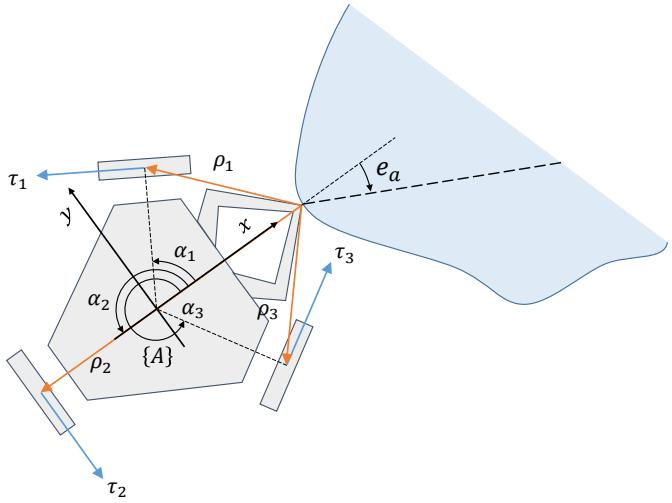

Fig. 3.3. Description of the parameters used for derivation of the control equations for robots

${}^{A}x$. Variables ρ_{wx} and ρ_{wy} are the x and y components of the vector from the tip of the end effector to the center of the wheel w. The above mapping can be used to write the robot Jacobian, J_A:

$$\begin{bmatrix} T_x & T_y & C_z \end{bmatrix}^T = J_A \begin{bmatrix} \tau_1 & \tau_2 & \tau_3 \end{bmatrix}^T. \tag{3.2}$$

Finally, the traction force for each wheel of the robot is calculated by substituting Eqn.(2.7) for T_x and T_y and solving Eqn.(3.2) for $[\tau_1, \tau_2, \tau_3]^T$. To correct the attitude of the robot towards the object, C_z is determined by applying a proportional controller with a gain K_a to the attitude error, e_a.

$$\vec{\tau} = J_A^{-1} \begin{bmatrix} {}^{A}_{O}R[\Omega(K_p e + K_d \dot{e})] \\ K_a e_a \end{bmatrix}, \tag{3.3}$$

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 orientation correcting moment.

3.1.3 Fabricated Prototype

The robot structure and wheels are fabricated by CO_2 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 control 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. 1.1. Each robot weighs 150 g and fits into a 127×117×50 mm box.

3.1.4 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 μm 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. 3.4. The experiments are conducted with two circular objects with masses of 100 and 230 g. The average coefficients

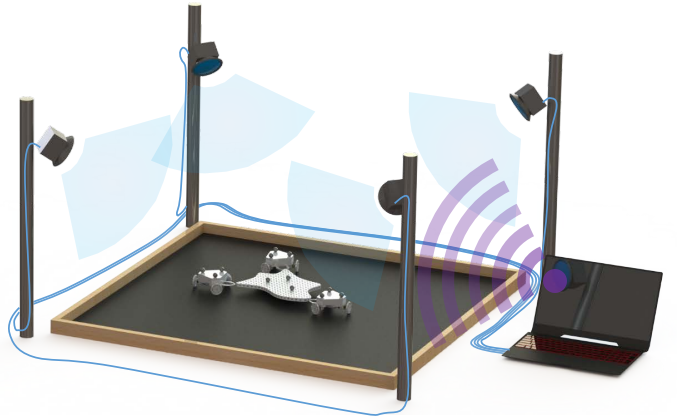

Fig. 3.4. An overview of the experimental setup used for validation of the proposed algorithm

of static and kinetic friction measured in different locations of the experimental environment are 0.44±0.1 and 0.22±0.07, respectively.

3.2 Results

This section explains the experimental setup and results of implementing the proposed decentralized collective manipulation algorithm on a real physical system.

One of the most important features of a swarm system is to demonstrate scalable, flexible, and robust system-level functionality [9]. Thus, several experimental scenarios are designed to study the system-level behavior and evaluate the efficiency of the proposed algorithm in different conditions. For instance, figure 3.5 illustrates snapshots of five $\Delta\rho$ 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 [1].

The experimental scenarios were tested with both decentralized and centralized controllers and responses of both control strategies are compared and tabulated for evaluating the efficiency and effectiveness of the proposed algorithm. A comparison

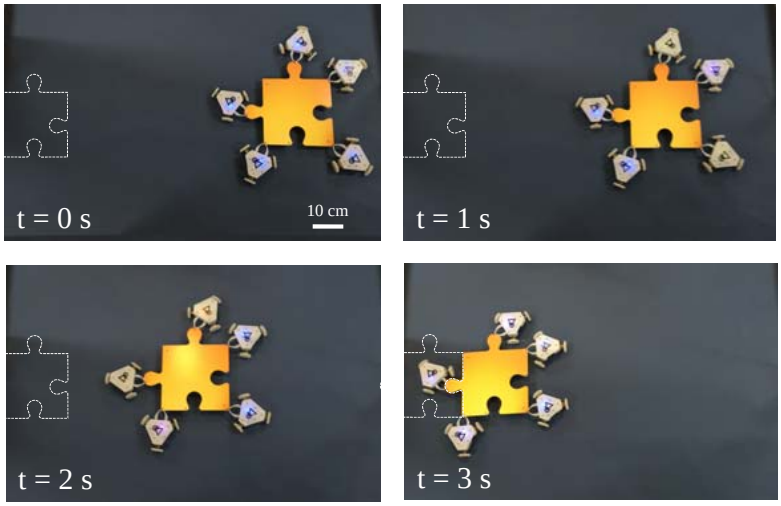

Fig. 3.5. Snapshots of five $\Delta\rho$ robots as they are manipulating a 100 g object. The manipulated object is assembled to a virtual puzzle piece depicted with white dashed line

between time responses of the object for centralized and decentralized controllers is depicted in Fig. 3.6.

3.2.1 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. 3.7, 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 orientation errors and shorter settling times for the system. This is expected due to an increase in the resultant forces applied to the CoM of the object based on the model presented in Eqn.(2.2). 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

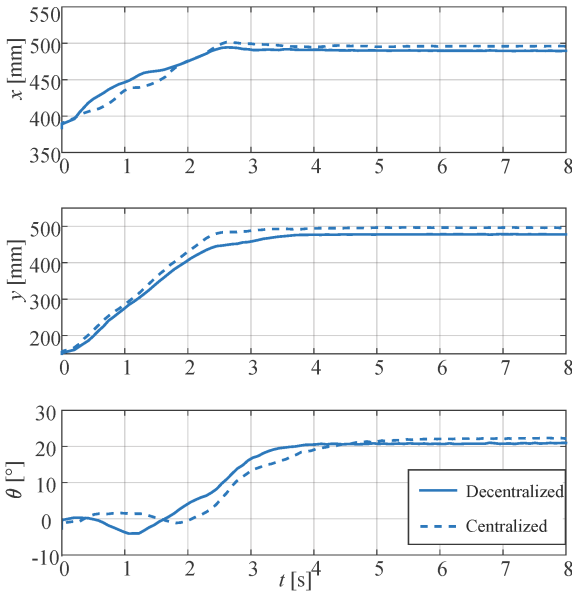

Fig. 3.6. Time responses of the system for decentralized and centralized controllers

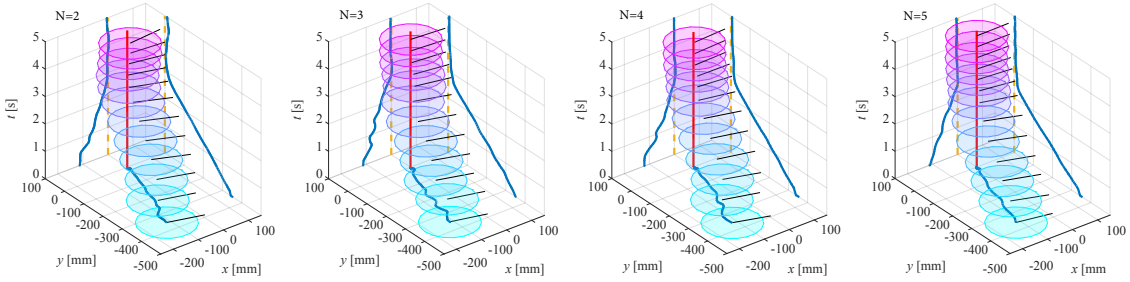

Fig. 3.7. Position and orientation of the object over time (vertical axis) for different group populations N. Path of the CoM of the object is projected on $x-y$ plane. Time responses for x and y are plotted in $y-t$ and $x-t$ planes respectively. The red line represents the location of the goal over time

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 equations Eqns.(2.15 and 2.16) in Section 2.3 are greater than the eigenvalues of the identity matrix $J[J^+]$ (which is equal to 1) for the centralized controller.

Table 3.1 presents the average settling times for manipulating the object. Throughout this manuscript, two settling time metrics are defined as the time elapsed from the start of the experiment to the time at which the CoM of the object 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 3.1.

Table. 3.1. 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±2.40 °	7.1±0.81 mm, 8.8±1.92 °
3	4.3±0.66 mm, 5.7±1.21 °	3.5±0.75 mm, 4.3±0.85 °
4	3±0.78 mm, 2.2±0.93 °	2.4±0.74 mm, 1.4±0.68 °
5	0.8±0.41 mm, 0.6 ±0.47 °	0.2±0.30 mm, 0.2±0.11 °

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. 3.7 and Table 3.1 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 previously studied this effect in [16] 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 conducted 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.

3.2.2 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.

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 $\Delta\rho$ 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 CoM of the object over time are displayed in Fig. 3.8-A. Table 3.2 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 research.

Table. 3.2. Settling time and steady-state error for different payloads M

Settling time				
	$e = 15\%$		$e = 5\%$	
M(Kg)	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

Steady-state error		
M (Kg)	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 °

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 arbi-

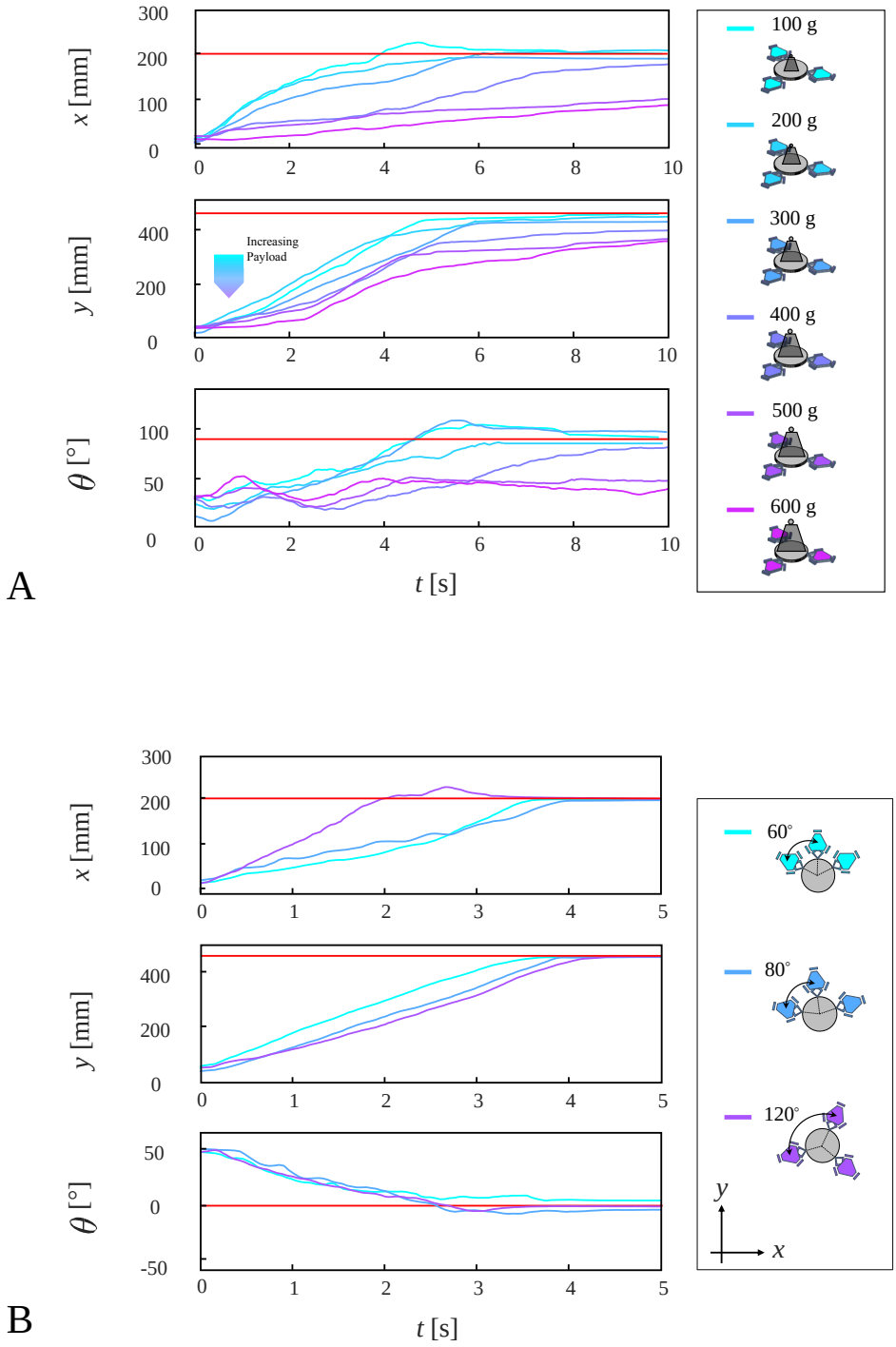

Fig. 3.8. The x, y positions and the orientation of the CoM of the object over time for (A) different payloads ranging from 100 g (cyan) to 600 g (magenta) and (B) different group formations (indicated by different colors). The red line represents the desired object pose

trary 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. 3.8-B. As shown in this figure, robots successfully move a 100 g object to the desired location and orientation for all of the considered formations. Table 3.3 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 [16]. 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 2.3.

3.2.3 Robustness

Robustness of the system is characterized by several factors [35, 40], which include: 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

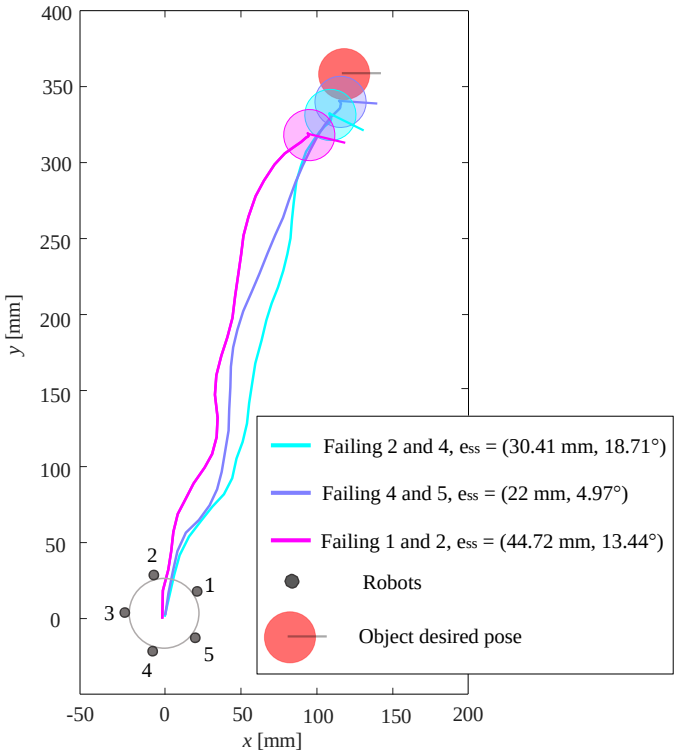

Fig. 3.9. System configuration for robustness test. Initial robot positions around the object are represented using small gray circles. The desired pose of the object is depicted with red. In each trial, different selection of the robots involved in the manipulation task are set to be inactive after 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

Table. 3.3. Settling time and steady-state error for different group formations

Settling time				
	$e = 15\%$		$e = 5\%$	
Increments	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

Steady-state error		
Increments	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 °

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. Compared to complex robots, simple agents are less likely to experience failures through the period of operation.

The experiments presented in this section are designed to investigate the redundancy of the system. In this regard, the agents are intentionally programmed to fail during the experiment. Figure 3.9 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 (friction forces added due to inactive agents that remain attached to the object through the manipulation phase).

3.2.4 Efficiency

In this study, 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).

Object velocity profile

The velocity profiles of the object during manipulation have been used in [4] as a method to determine the coordination level of the system. As suggested in [4], 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.

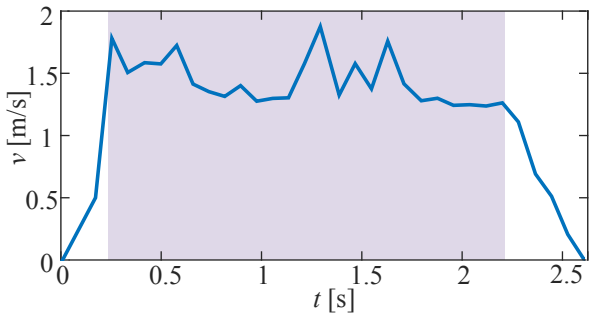

Fig. 3.10. Velocity profile of the object during the manipulation

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 manuscript.

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 [22,46]. 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. 3.10 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

Table. 3.4. 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

$RMSD$		
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

the object, 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:

$$RMSD(v) = \sqrt{\frac{\sum_{i=1}^{n}(\hat{v} - v_i)^2}{n}}, \tag{3.4}$$

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 [2, 25, 34, 52].

Table 3.4 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.

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 [21]. Similar to the analogy used for velocity profiles, path efficiency is a performance index, which shows the amount of deviation from the shortest path and is defined as:

$$\eta_{path}(s) = \frac{\|\vec{x}_n - \vec{x}_0\|}{\sum_{i=0}^{n-1} \|\vec{x}_{i+1} - \vec{x}_i\|}, \tag{3.5}$$

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 3.5 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 3.5, 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 3.5 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.

Table. 3.5. 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

Path Efficiency		
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

3.2.5 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 3.11. Although this experiment shows the effectiveness of the algorithm in trajectory tracking, further extensive investigations are required to evaluate its performance, which is also the subject of future work.

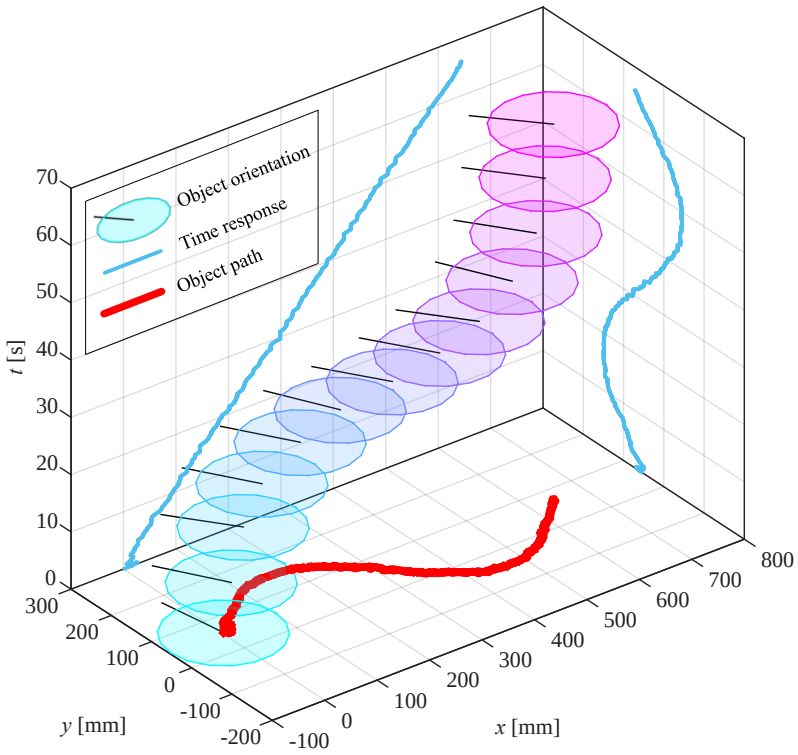

Fig. 3.11. Timelapse figure of position and orientation of the object being moved along a sinusoidal trajectory. Path of the CoM of the object is projected on $x - y$ plane in red. Time responses for x and y are plotted in $y - t$ and $x - t$ planes respectively

Chapter 4

Collective Impedance Control

Although cooperative manipulation is achievable through basic algorithms (such as centralized pose control of all the agents surrounding the object), complete control on the trajectory of the manipulated object and its interactions with the environment requires more advanced approaches. A possible approach to control the response of the object in regards to external forces and reactions from the environment is to utilize an impedance control framework [24], which also provides the possibility to control both the absolute motion and the internal forces of the object (e.g. manipulation of soft and deformable objects [32]). Moreover, impedance control provides a safer environment for scenarios which involve human cooperation or physical interaction of objects with each other (e.g. collective construction).

4.1 Introduction

The implementation of impedance control in multi-robot systems is commonly achieved via two distinct ways: 1) defining the input force vector such that the response of the system converges to the response of the desired impedance dynamics (e.g. replacing the uncontrolled dynamics with the desired system response via feedback-

linearization) [7,23,41]; and 2) pose control of the object with respect to the desired impedance dynamics [8].

The formulation presented in this section focuses on the derivation of a scalable algorithm to directly control the forces applied to an object in order to follow desired impedance dynamics based on our earlier algorithm. Although derivation of the feedback-linearized forces to substitute the dynamics follows the same approach presented in [7,14,26,32,41], unlike these previous methods, the proposed algorithm does not use any information about the team formation and population. Consequently, the real Jacobian of the system (or the corresponding grasp matrix [33], which is constructed based on the number of manipulators involved in the task and their end-point position) is unknown to the system.

After calculating the required total force vector in response to the applied external force, a simple and not necessarily accurate assumption is made to distribute the force vector among the team members. Through the course of manipulation, each robot assumes there is only one virtual teammate, which is located at its mirror position with respect to the Center of Mass (CoM) of the object. Although the "virtual agents" may not necessarily represent the effects of the team, the virtual cooperation between each robot and its corresponding virtual teammate leads to a stable and effective distribution of the defined force vector among the agents.

One of the main challenges to achieving impedance control with this approach is caused due to the scaling of the force vector by the number of agents. Although this effect might be desirable in some applications, it causes the system to deviate from the desired impedance behavior. This problem is solved in this work using a gradient decent optimization algorithm that estimates a scaling parameter (which represents the team population) on the control equation in order to minimize deviations from the desired response.

Simulation results of the system with different group populations and formations demonstrate the effectiveness of the algorithm in both estimating the team population and providing the desired impedance response. In addition to scalability, this framework eliminates the need for inter-agent communication (since it does not require any knowledge on the team population or formation). Moreover, the force distribution algorithm and the population estimating methods proposed in this research can be utilized to generate any other desired system response without modification.

The notations used in this section follows the style presented before. Based on this notation, ${}^{i}\bar{v}$ denotes a vector $\bar{v}$, which is defined in coordinate frame {i} and a transformation $\mathbf{A}$ form coordinate frame {i} to {j} is denoted by ${}^{\mathrm{j}}_{\mathrm{i}}\mathbf{A}$. Furthermore, throughout this manuscript, all matrices are indicated with bold non-italic symbols while vectors are denoted with an overbar and are single column. The phrase "agent position" refers to the point in which the agent applies forces to the object and is defined in the object's coordinate frame {O}. Similarly, the phrase "virtual agent position" refers to the point defined in {O} where a virtual agent applies an imaginary force to the object (note that since this force is not real, it does not contribute to the system dynamics). The transpose operation on a given matrix $\mathbf{A}$ is noted with $\mathbf{A}^T$ and the Moore-Penrose pseudoinverse of $\mathbf{A}$ is noted with $\mathbf{A}^+$. Finally, $\mathbf{I}_n$ and $\mathbf{0}_n$ are used to represent $n \times n$ identity and zero matrices, respectively. The following subsections cover the details of the mathematical model of the system, the contact model, and the derivation of the decentralized impedance control equation.

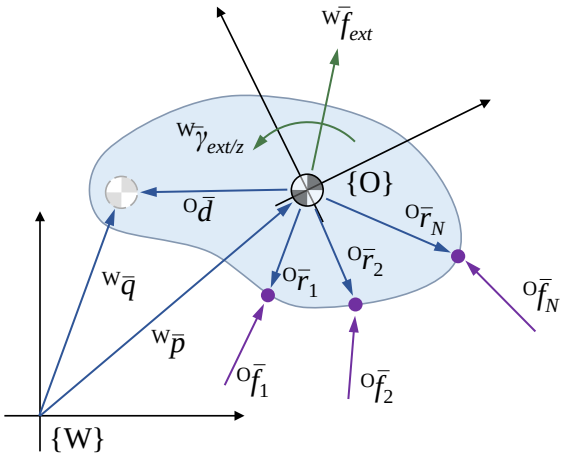

Fig. 4.1. Free body diagram of the system and the corresponding parameters. As depicted in this figure, coordinate frame {O} is attached to the real CoM of the object and its position is defined by vector ${}^{W}\bar{x}$. Vector ${}^{W}\bar{y}$, defined in the inertial frame {W}, points to the position of the apparent (desired) CoM of the impedance dynamics. Vectors ${}^{O}\bar{f}_1$ to ${}^{O}\bar{f}_N$ represent the forces applied by the agents involved in the manipulation which are located at ${}^{O}\bar{r}_1$ to ${}^{O}\bar{r}_N$ with respect to the CoM of the object. ${}^{W}\bar{f}_{ext}$ and ${}^{W}\bar{\gamma}_{ext/z}$ are the force vector and moment about the ${}^{W}\hat{k}$ axis that are caused by interactions of the object with the environment.

4.2 Modeling

In order to focus on the performance of the proposed controller and eliminating the complexities of 3-D dynamics, this manuscript covers the derivation of control equations for a rigid object which is constrained to planar motion and carried by a team of N agents. Figure 4.1 illustrates the free body diagram of the system and the corresponding parameters which are used in the mathematical model. As depicted in this figure, the non-inertial coordinate frame {O} is attached to the center of mass (CoM) of the object. Vectors ${}^{O}\bar{r}_i = [r_{xi}, r_{yi}]^T$ and ${}^{O}\bar{f}_i = [f_{xi}, f_{yi}]^T$ represent the position of the i^{th} agent with respect to the CoM of the object and the force vector applied by the corresponding agent to the object, respectively. It is also assumed that, although there is no constraint on the direction of ${}^{O}\bar{f}_i$ vectors, the agents do not apply direct moments to the object. This assumption is made based on the available

connection mechanisms on low-cost robots which include simple mechanical grippers (that act as a revolute joint) or electromagnetic attachments. Furthermore, this assumption increases the practicality of the algorithm by simplifying the attachment mechanism required by the agents.

Differential equation of motion of the object in the world coordinate frame {W} is defined as:

$$^{\mathrm{W}}\ddot{\bar{p}} = \mathbf{M}_o^{-1}\left({}^{\mathrm{W}}_{\mathrm{O}}\mathbf{R}\ {}^{\mathrm{O}}\bar{F} + {}^{\mathrm{W}}\bar{F}_{ext}\right), \tag{4.1}$$

where ${}^{\mathrm{W}}_{\mathrm{O}}\mathbf{R}$ is a rotation matrix that maps coordinate frame {O} to {W}. Vector $^{\mathrm{W}}\bar{p} = p_x \cdot {}^{\mathrm{W}}\hat{i} + p_y \cdot {}^{\mathrm{W}}\hat{j} + p_\theta \cdot {}^{\mathrm{W}}\hat{k}$ is the position of the CoM of the object. The matrix $\mathbf{M}_o$ is the mass matrix of the object and defined as:

$$\mathbf{M}_o = \begin{bmatrix} m_o & 0 & 0 \\ 0 & m_o & 0 \\ 0 & 0 & I_o \end{bmatrix}. \tag{4.2}$$

The force vector $^{\mathrm{O}}\bar{F}$ represents the resultant force applied to the object by all the agents involved in the manipulation task and $^{\mathrm{W}}\bar{F}_{ext}$ is the resultant external force vector due to interactions of the object with the environment. Both $^{\mathrm{W}}\bar{F}_{ext}$ and $^{\mathrm{O}}\bar{F}$ include the linear forces along x and y directions and moments about z axis:

$$^{\mathrm{W}}\bar{F}_{ext} = f_{ext/x} \cdot {}^{\mathrm{W}}\hat{i} + f_{ext/y} \cdot {}^{\mathrm{W}}\hat{j} + \gamma_{ext/z} \cdot {}^{\mathrm{W}}\hat{k}, \tag{4.3}$$

$$^{\mathrm{O}}\bar{F} = f_x \cdot {}^{\mathrm{O}}\hat{i} + f_y \cdot {}^{\mathrm{O}}\hat{j} + \gamma_z \cdot {}^{\mathrm{O}}\hat{k} \tag{4.4}$$

$$= \mathbf{J}\begin{bmatrix} {}^{\mathrm{O}}\bar{f}_1{}^T & \dots & {}^{\mathrm{O}}\bar{f}_N{}^T \end{bmatrix}^T, \tag{4.5}$$

where $\mathbf{J}$ is the real Jacobian of the system and defined as:

$$\mathbf{J} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{I}_2 & \cdots & \mathbf{I}_2 \\ {}^{\mathrm{O}}\bar{r}_1{}^T & {}^{\mathrm{O}}\bar{r}_2{}^T & \cdots & {}^{\mathrm{O}}\bar{r}_N{}^T \end{bmatrix}. \tag{4.6}$$

4.2.1 Controller formulation

The goal is to formulate a control equation for each agent that is independent of the team formation and population and provides the desired impedance response of an object that is manipulated by a group of robots and possibly humans. To do so, the decentralized force controller derivation is modified to yield the desired impedance dynamics. Additionally, a gradient descent optimization algorithm is formulated to minimize the error between the desired and actual system responses.

In this formulation, each agent solely relies on a local Jacobian to modulate its actions and forces while collaborating with the others. The local Jacobians are constructed to model the behavior of the teammates with a single virtual agent without knowledge of the real team formation and population. Consequently, the proposed control law is scalable to teams with any population and formation.

The derivation of the control equation starts with the construction of the local Jacobian for the i^{th} agent:

$$\mathbf{J}_i = \begin{bmatrix} \mathbf{I}_2 & \mathbf{I}_2 \\ {}^{\mathrm{O}}\bar{r}_i^T & {}^{\mathrm{O}}\bar{a}_i^T \end{bmatrix}, \tag{4.7}$$

where ${}^{\mathrm{O}}\bar{a}_i = [a_{xi},\ a_{yi}]^T$ represents the position of the virtual agent for the i^{th} agent in $\{\mathrm{O}\}$. After constructing the local Jacobian, its Moore-Penrose pseudoinverse can be used to distribute the desired force vector, ${}^{\mathrm{O}}\bar{F}^*$, for each team member as:

$${}^{\mathrm{O}}\bar{f}_i = \begin{bmatrix} \mathbf{I}_2 & \mathbf{0}_2 \end{bmatrix} \mathbf{J}_i^+ \, {}^{\mathrm{O}}\bar{F}^*, \quad \forall i = 1, 2, \cdots, N. \tag{4.8}$$

In the above equation, ${}^{\mathrm{O}}\bar{f}_i$ represents the force vector which is applied to the object by the i^{th} agent. Since this force distribution depends on the assumed positions of the virtual agents, as shown in section 2.1, substituting ${}^{\mathrm{O}}\bar{a}_i$ with $-{}^{\mathrm{O}}\bar{r}_i$ results in a stable mapping between ${}^{\mathrm{O}}\bar{F}$ and ${}^{\mathrm{O}}\bar{F}^*$ vectors. In other words, defining ${}^{\mathrm{O}}\bar{a}_i = -{}^{\mathrm{O}}\bar{r}_i$ results in a positive inner product of ${}^{\mathrm{O}}\bar{F}^*$ and the resultant force applied by all the agents:

$$\langle {}^{\mathrm{O}}\bar{F}^* \ , \ \mathbf{J} \begin{bmatrix} {}^{\mathrm{O}}\bar{f}_1{}^T & \dots & {}^{\mathrm{O}}\bar{f}_N{}^T \end{bmatrix}^T \rangle > 0, \tag{4.9}$$

The final decentralized control equation is achieved by calculating $\bar{F}^*$ in (4.8) based on the desired impedance dynamics which is defined as:

$$\mathbf{M}_d \ {}^{\mathrm{W}}\ddot{\bar{q}}_d + \mathbf{B}_d \ \Delta^{\mathrm{W}}\dot{\bar{q}} + \mathbf{K}_d \ \Delta^{\mathrm{W}}\bar{q} = {}^{\mathrm{W}}\bar{F}_{ext}, \tag{4.10}$$

where $\mathbf{M}_d$, $\mathbf{B}_d$ and $\mathbf{K}_d$ are 3×3 positive definite matrices of desired mass, damping and stiffness of the impedance dynamics. Vectors $\Delta^{\mathrm{W}}\dot{\bar{q}}$ and $\Delta^{\mathrm{W}}\bar{q}$ are equal to:

$$\begin{aligned} \Delta^{\mathrm{W}}\bar{q} &= \ {}^{\mathrm{W}}\bar{q} - \ {}^{\mathrm{W}}\bar{q}_0 \\ \Delta^{\mathrm{W}}\dot{\bar{q}} &= \ {}^{\mathrm{W}}\dot{\bar{q}} - \ {}^{\mathrm{W}}\dot{\bar{q}}_0 \end{aligned} \tag{4.11}$$

Vectors ${}^{\mathrm{W}}\hat{q}$ and ${}^{\mathrm{W}}\dot{\hat{q}}$ define the location and velocity of the apparent (desired) CoM of the system, respectively. ${}^{\mathrm{W}}\hat{q}_0$ represent the position where the potential energy of the desired impedance dynamics is zero and ${}^{\mathrm{W}}\dot{\hat{q}}_0$ is the zero velocity for the viscous damping term. The relation between ${}^{\mathrm{W}}\hat{q}$ and its corresponding derivatives with ${}^{\mathrm{W}}\bar{p}$,

${}^{\mathrm{W}}\dot{\bar{p}}$ and ${}^{\mathrm{W}}\ddot{\bar{p}}$ are:

$$\begin{aligned}
{}^{\mathrm{W}}\bar{q} &= {}^{\mathrm{W}}\bar{p} + {}^{\mathrm{W}}_{\mathrm{O}}\mathbf{R}\ {}^{\mathrm{O}}\bar{d} \\
{}^{\mathrm{W}}\dot{\bar{q}} &= {}^{\mathrm{W}}\dot{\bar{p}} + {}^{\mathrm{W}}_{\mathrm{O}}\mathbf{R}({}^{\mathrm{O}}\bar{\omega}_o \times {}^{\mathrm{O}}\bar{d}) \\
{}^{\mathrm{W}}\ddot{\bar{q}} &= {}^{\mathrm{W}}\ddot{\bar{p}} + {}^{\mathrm{W}}_{\mathrm{O}}\mathbf{R}({}^{\mathrm{O}}\dot{\bar{\omega}}_o \times {}^{\mathrm{O}}\bar{d} + {}^{\mathrm{O}}\bar{\omega}_o \times {}^{\mathrm{O}}\bar{\omega}_o \times {}^{\mathrm{O}}\bar{d}).
\end{aligned} \tag{4.12}$$

Since ${}^{\mathrm{O}}\dot{\bar{\omega}}_o = {}^{\mathrm{O}}\ddot{\bar{p}} \cdot {}^{\mathrm{O}}\hat{k} = \ddot{\theta}_o$ and ${}^{\mathrm{W}}_{\mathrm{O}}\mathbf{R} = \mathbf{R}_z(\theta_o)$ is the rotation about ${}^{\mathrm{W}}\hat{k}$, the acceleration of the apparent CoM of the object, ${}^{\mathrm{W}}\ddot{\bar{q}}$, can be written as:

$${}^{\mathrm{W}}\ddot{\bar{q}} = \mathbf{H}\ {}^{\mathrm{W}}\ddot{\bar{p}} + {}^{\mathrm{W}}_{\mathrm{O}}\mathbf{R}({}^{\mathrm{O}}\bar{\omega}_o \times {}^{\mathrm{O}}\bar{\omega}_o \times {}^{\mathrm{O}}\bar{d}), \tag{4.13}$$

where the 3×3 matrix $\mathbf{H}$ is obtained by solving $\mathbf{H}\ {}^{\mathrm{W}}\ddot{\bar{p}} = {}^{\mathrm{W}}\ddot{\bar{p}} + {}^{\mathrm{W}}_{\mathrm{O}}\mathbf{R}({}^{\mathrm{O}}\dot{\bar{\omega}}_o \times\ {}^{\mathrm{O}}\bar{d})$ and defined as:

$$\mathbf{H} = \mathbf{I}_3 + \frac{\partial}{\partial({}^{\mathrm{W}}\ddot{\bar{p}} \cdot \hat{w})}\Big({}^{\mathrm{W}}_{\mathrm{O}}\mathbf{R}({}^{\mathrm{O}}\dot{\bar{\omega}}_o \times\ {}^{\mathrm{O}}\bar{d})\Big), \quad w = \{i,\ j,\ k\} \tag{4.14}$$

Solving (4.13) for ${}^{\mathrm{W}}\ddot{\bar{p}}$ and substituting the results in (4.1) leads to an expression for ${}^{\mathrm{W}}\ddot{\bar{q}}$ as a function of ${}^{\mathrm{W}}\bar{F}$ and ${}^{\mathrm{W}}\bar{F}_{ext}$. Similarly, an expression for the desired acceleration of the apparent CoM of the object can be obtained by solving (4.10) for ${}^{\mathrm{W}}\ddot{\bar{q}}_d$. The corresponding expressions are presented in what follows.

$${}^{\mathrm{W}}\ddot{\bar{q}} = \mathbf{H}\mathbf{M}_o^{-1}({}^{\mathrm{W}}\bar{F} + {}^{\mathrm{W}}\bar{F}_{ext}) + {}^{\mathrm{W}}\bar{C}, \tag{4.15}$$

$${}^{\mathrm{W}}\ddot{\bar{q}}_d = \mathbf{M}_d^{-1}({}^{\mathrm{W}}\bar{F}_{ext} -\ {}^{\mathrm{W}}\bar{F}_{imp}), \tag{4.16}$$

where ${}^{\mathrm{W}}\bar{C}$ and ${}^{\mathrm{W}}\bar{F}_{imp}$ are defined as:

$$ {}^{\mathrm{W}}\bar{C} = {}^{\mathrm{W}}_{\mathrm{O}}\mathbf{R}({}^{\mathrm{O}}\bar{\omega}_o \times {}^{\mathrm{O}}\,\bar{\omega}_o \times \ {}^{\mathrm{O}}\bar{d}), \tag{4.17} $$

$$ {}^{\mathrm{W}}\bar{F}_{imp} = \mathbf{B}_d\ \Delta^{\mathrm{W}}\dot{\bar{q}} + \mathbf{K}_d\ \Delta^{\mathrm{W}}\bar{q}. \tag{4.18} $$

Finally, the desired control force,${}^{\mathrm{W}}\bar{F}^*$, for the given impedance dynamics can be computed by equating the desired and real accelerations of the apparent CoM of the object (i.e. ${}^{\mathrm{W}}\ddot{\bar{q}}_d = {}^{\mathrm{W}}\ddot{\bar{q}}$) and solving for ${}^{\mathrm{W}}\bar{F}$. Consequently, the overall control equation for each agent is obtained as:

$$ \bar{f}_i = \begin{bmatrix} \mathbf{I}_2 & \mathbf{0}_2 \end{bmatrix} \mathbf{J}_i^{+}\ {}^{\mathrm{O}}_{\mathrm{W}}\mathbf{R}\ {}^{\mathrm{W}}\bar{F}^*, \tag{4.19} $$

where ${}^{\mathrm{W}}\bar{F}^*$ defined as:

$$ \begin{aligned} {}^{\mathrm{W}}\bar{F}^* &= {}^{\mathrm{W}}\bar{F}^*_{ext} - {}^{\mathrm{W}}\bar{F}^*_{imp}, \\ {}^{\mathrm{W}}\bar{F}^*_{ext} &= \left(\mathbf{M}_o\mathbf{H}^{-1}\mathbf{M}_d^{-1} - \mathbf{I}_3\right)\ {}^{\mathrm{W}}\bar{F}_{ext}, \\ {}^{\mathrm{W}}\bar{F}^*_{imp} &= \mathbf{M}_o\mathbf{H}^{-1}\left(\mathbf{M}_d^{-1}\ {}^{\mathrm{W}}\bar{F}_{imp} + {}^{\mathrm{W}}\bar{C}\right). \end{aligned} \tag{4.20} $$

Equation (4.19) along with (4.20) describe the decentralized impedance control equation for each agent. Note that the derived control law does not include any information about the team formation and population and is scalable to any team with $N \geq 2$.

To further analyze the performance of the controller, it is essential to understand the relation between ${}^{\mathrm{O}}\bar{F}^*$ and ${}^{\mathrm{O}}\bar{F}$. Based on (4.5) the resultant force applied to the object is equal to the real Jacobian of the system multiplied by a force vector that includes all the agent forces. Thus, the mapping between ${}^{\mathrm{O}}\bar{F}^*$ and ${}^{\mathrm{O}}\bar{F}$ is obtained

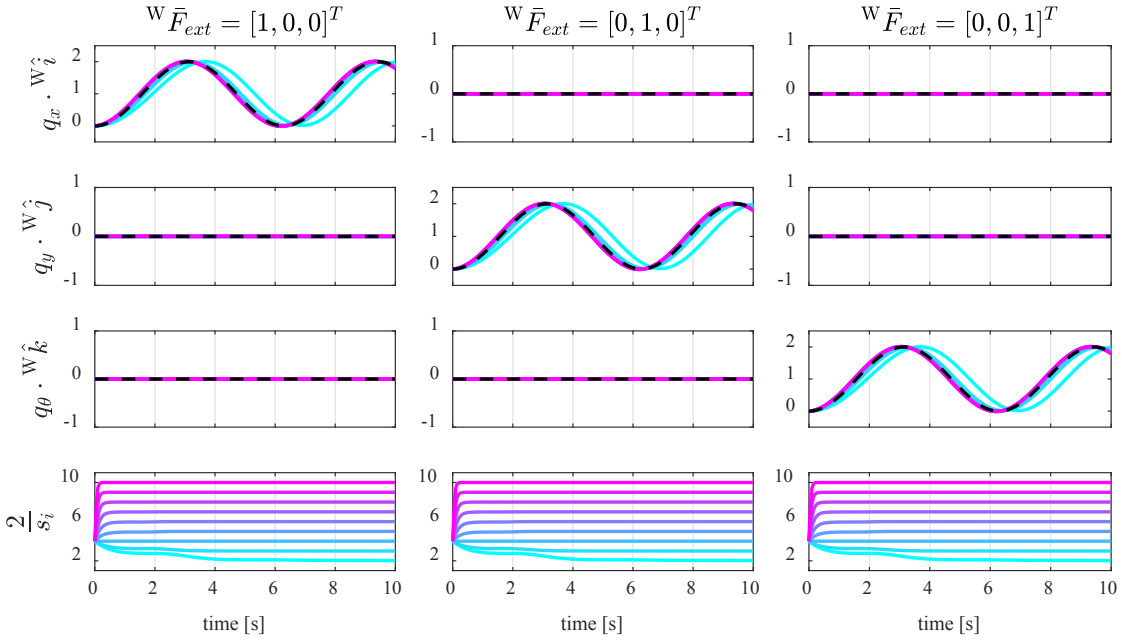

Fig. 4.2. The effect of team population on the response of the system. The line colors are used to indicate the population of the team which changes from 2 (cyan) to 10 (magenta). The team members are uniformly distributed around the object. Each column shows the response of the system to a step external force along ${}^{W}\hat{i}$, ${}^{W}\hat{j}$ and ${}^{W}\hat{k}$ axes, respectively. While the first three rows depict the time response of the system, the last row shows convergence of the scaling parameter, s_i. The response of the desired impedance dynamics to the external forces is depicted with dashed black lines.

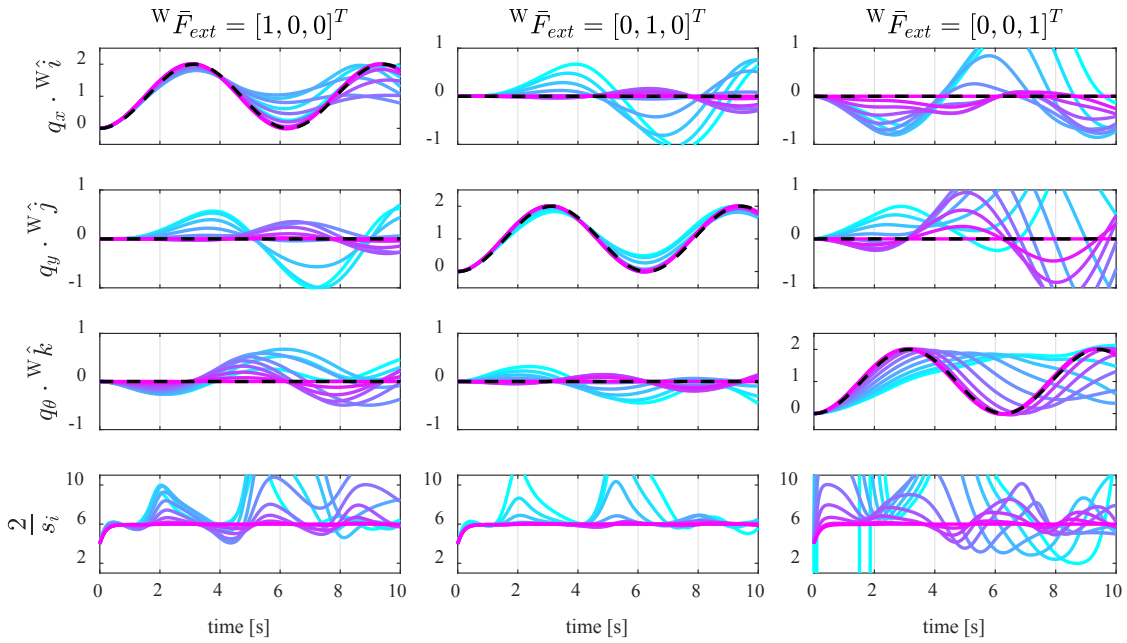

Fig. 4.3. The effect of team formations on the response of the system. The line colors are used to indicate different team formations. The initial response, depicted with cyan color, corresponds to a formation in which all the 6 agents are uniformly distributed in a quadrant of the object. The region in which the agents are distributed is gradually increased to the full circle (indicated with magenta color) with 30 degree increments. Each column shows the response of the system to a step external force along ${}^{\mathrm{W}}\hat{i}$, ${}^{\mathrm{W}}\hat{j}$ and ${}^{\mathrm{W}}\hat{k}$ axes, respectively. While the first three rows depict the time response of the system, the last row shows convergence of the scaling parameter, s_i. The response of the desired impedance dynamics to the external forces is depicted with dashed black lines.

by substituting $\bar{f}_i$ vectors in (4.5) with their definitions presented in (4.8):

$$ {}^{\mathrm{O}}\bar{F} = \mathbf{J} \begin{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \mathbf{0}_2 \end{bmatrix} \mathbf{J}_1^+ \\ \vdots \\ \begin{bmatrix} \mathbf{I}_2 & \mathbf{0}_2 \end{bmatrix} \mathbf{J}_N^+ \end{bmatrix} {}^{\mathrm{O}}\bar{F}^* = \mathbf{T}\ {}^{\mathrm{O}}\bar{F}^*, \tag{4.21} $$

where $\mathbf{T}$ is a 3×3 transformation matrix that maps ${}^{\mathrm{O}}\bar{F}^*$ to ${}^{\mathrm{O}}\bar{F}$. Utilizing the assumptions made on the location of the virtual agents (i.e. ${}^{\mathrm{O}}\bar{a}_i = -{}^{\mathrm{O}}\bar{r}_i$), the transformation matrix, $\mathbf{T}$, simplifies to:

$$ \mathbf{T} = \frac{1}{2} \begin{bmatrix} N & 0 & -\sum_{i=1}^{N} \frac{{}^{\mathrm{O}}r_{yi}}{\|{}^{\mathrm{O}}r_i\|^2} \\ 0 & N & -\sum_{i=1}^{N} \frac{{}^{\mathrm{O}}r_{xi}}{\|{}^{\mathrm{O}}r_i\|^2} \\ \sum_{i=1}^{N} {}^{\mathrm{O}}r_{yi} & \sum_{i=1}^{N} {}^{\mathrm{O}}r_{xi} & N \end{bmatrix}. \tag{4.22} $$

In order to distinguish between the uncoupled and coupled mappings, the transformation matrix $\mathbf{T}$ can be written as a summation of two separate matrices: $\mathbf{T} = \mathbf{T}_u + \mathbf{T}_c$. Thus the mapping presented in (4.21) can be rewritten as:

$$ {}^{\mathrm{O}}\bar{F} = \mathbf{T}\ {}^{\mathrm{O}}\bar{F}^* = \mathbf{T}_u\ {}^{\mathrm{O}}\bar{F}^* + \mathbf{T}_c\ {}^{\mathrm{O}}\bar{F}^*, \tag{4.23} $$

where $\mathbf{T}_u$ and $\mathbf{T}_c$ are equal to:

$$ \mathbf{T}_u = \frac{N}{2}\mathbf{I}_3 \quad \mathbf{T}_c = \mathbf{T} - \mathbf{T}_u \tag{4.24} $$

Thus, as observed in (4.23), in the absence of coupling forces and moments, the forces applied to the object are scaled with the factor of $N/2$. Although this effect is desirable in simple manipulation tasks, the scaled value of ${}^{\mathrm{O}}\bar{F}^*$ causes the response

of the system to deviate from the desired impedance dynamics. In order to address this concern, a scaling parameter, s_i, is introduced to the control equation presented in (4.19) which modifies the corresponding equation to:

$$\bar{f}_i = \begin{bmatrix} \mathbf{I}_2 & \mathbf{0}_2 \end{bmatrix} \mathbf{J}_i^{+} \, {}^{\mathrm{O}}_{\mathrm{W}}\mathbf{R} \, {}^{\mathrm{W}}\bar{F}^{*} s_i, \tag{4.25}$$

In order to keep the control algorithm independent of the team population, the value of s_i can be defined by minimizing the error between ${}^{\mathrm{W}}\ddot{\bar{q}}$ and ${}^{\mathrm{W}}\ddot{\bar{q}}_d$. Note that, if there is any information regarding the team population, then s_i for each agent can be simply assigned to be $2/N$, which compensates for the scaling effect of the $\mathbf{T}$ matrix.

The following optimization problem is formulated to determine the values of scaling factors s_i for each agent:

$$\underset{s_i \in \mathbb{R}}{\text{minimize}} \quad \| {}^{\mathrm{W}}\ddot{\bar{q}}_d - {}^{\mathrm{W}}\ddot{\bar{q}} \|^2 \tag{4.26}$$

Since the objective function is convex and the feasible set is both closed and convex, the problem has only one global solution. Thus, the corresponding optimization problem can be solved with the gradient decent method where the update function for s_i is defined as:

$$(s_i)_{k+1} = (s_i)_k - \gamma \nabla \big(\| {}^{\mathrm{W}}\ddot{\bar{q}}_d - {}^{\mathrm{W}}\ddot{\bar{q}} \|^2 \big), \; k \geq 0 \tag{4.27}$$

The fact that s_i is a scalar parameter, reduces $\nabla \big(\| {}^{\mathrm{W}}\ddot{\bar{q}}_d - {}^{\mathrm{W}}\ddot{\bar{q}} \|^2 \big)$ to partial derivative of $\| {}^{\mathrm{W}}\ddot{\bar{q}}_d - {}^{\mathrm{W}}\ddot{\bar{q}} \|^2$ with respect to s_i. Moreover, since ${}^{\mathrm{W}}\ddot{\bar{q}}_d$ is not a function of s_i,

the corresponding derivative simplifies to:

$$\frac{\partial}{\partial s_i}(\|{}^{\mathrm{W}}\ddot{\bar{q}}_d - {}^{\mathrm{W}}\ddot{\bar{q}}\|^2) = -2({}^{\mathrm{W}}\ddot{\bar{q}}_d - {}^{\mathrm{W}}\ddot{\bar{q}})^T \frac{\partial}{\partial s_i}({}^{\mathrm{W}}\ddot{\bar{q}}) \tag{4.28}$$

The final step in deriving the update rule for s_i values is to find the partial derivative of ${}^{\mathrm{W}}\ddot{\bar{q}}$ with respect to s_i as presented below. Substituting (4.21) in (4.15) yields:

$${}^{\mathrm{W}}\ddot{\bar{q}} = \mathbf{H}\mathbf{M}_o^{-1\mathrm{W}}\bar{F}_{ext} + {}^{\mathrm{W}}\bar{C} + \mathbf{H}\mathbf{M}_o^{-1}\mathbf{T}\ {}^{\mathrm{W}}\bar{F}^* s_i \tag{4.29}$$

Taking partial derivative of (4.29) with respect to s_i yields:

$$\frac{\partial}{\partial s_i}({}^{\mathrm{W}}\ddot{\bar{q}}) = \mathbf{H}\mathbf{M}_o^{-1}\mathbf{T}\ {}^{\mathrm{W}}\bar{F}^* \tag{4.30}$$

Note that, since the matrix $\mathbf{T}$, which is a function of the real Jacobian of the system is unknown, it needs to be eliminated from the formulation. In this regard, it is possible to substitute $\mathbf{H}\mathbf{M}_o^{-1}\mathbf{T}\ {}^{\mathrm{W}}\bar{F}^*$ in (4.29) with its expression in (4.30) and solve for $\partial\ {}^{\mathrm{W}}\ddot{\bar{q}}/\partial s_i$ which yields:

$$\frac{\partial}{\partial s_i}({}^{\mathrm{W}}\ddot{\bar{q}}) = \frac{1}{s}({}^{\mathrm{W}}\ddot{\bar{q}} - \mathbf{H}\mathbf{M}_o^{-1\mathrm{W}}\bar{F}_{ext} - {}^{\mathrm{W}}\bar{C}). \tag{4.31}$$

Finally, the update rule for the scaling parameter of each agent, s_i, is obtained by substituting (4.31) in (4.27):

$$(s_i)_{k+1} = (s_i)_k + \frac{2\gamma}{(s_i)_k}({}^{\mathrm{W}}\ddot{\bar{q}}_d - {}^{\mathrm{W}}\ddot{\bar{q}})^T({}^{\mathrm{W}}\ddot{\bar{q}} - \mathbf{H}\mathbf{M}_o^{-1\mathrm{W}}\bar{F}_{ext} - {}^{\mathrm{W}}\bar{C}) \tag{4.32}$$

4.3 Simulations and Results

The response and performance of the proposed decentralized impedance control architecture is tested through various simulated scenarios. The details of each simulation and the corresponding results are presented in what follows. In each scenario, the response of the decentralized impedance controller is compared with the desired impedance dynamics.The corresponding simulations and visualizations presented in this manuscript are performed with MATLAB software.

4.3.1 System response to an external force

This section focuses on a set of simulations that are performed to study the effects of team population and formation on the time response of the system as well as the effectiveness of the optimization algorithm in detecting the team population. The corresponding results of the simulations are depicted in Fig. 4.2 and Fig. 4.3. The parameters that are used for these simulations are: ${}^{\mathrm{O}}\bar{d} = [0, 0, 0]^T$, $m_o = 10$ kg, $I_o = 10$ kg.m^2, $\mathbf{M}_d = \mathbf{I}_3$ {kg, kg.m^2}, $\mathbf{B}_d = \mathbf{0}_3$, and $\mathbf{K}_d = \mathbf{I}_3$ {N/m, Nm/rad}. All the initial conditions are set to be zero (i.e. ${}^{\mathrm{W}}\bar{p}_0 = {}^{\mathrm{W}}\dot{\bar{p}}_0 = [0, 0, 0]^T$) and both ${}^{\mathrm{W}}\bar{q}_0$ and ${}^{\mathrm{W}}\dot{\bar{q}}_0$ are set to be zero.

The results depicted in Fig. 4.2 show the time response of the system with different team populations that uniformly enclose a circular object, to external forces along ${}^{\mathrm{W}}\hat{i}$, ${}^{\mathrm{W}}\hat{j}$ axes, and an external moment about ${}^{\mathrm{W}}\hat{k}$ axis. As depicted in this figure, the formulated optimization problem causes s_i to converge to $2/N$ and thus, it can successfully estimate the number of agents involved in the task. Also, as the population of the team increases, s_i demonstrates a faster convergence and consequently, system dynamics converges to the desired impedance dynamics.

Similarly, Fig. 4.3 illustrates the response of the system with different team

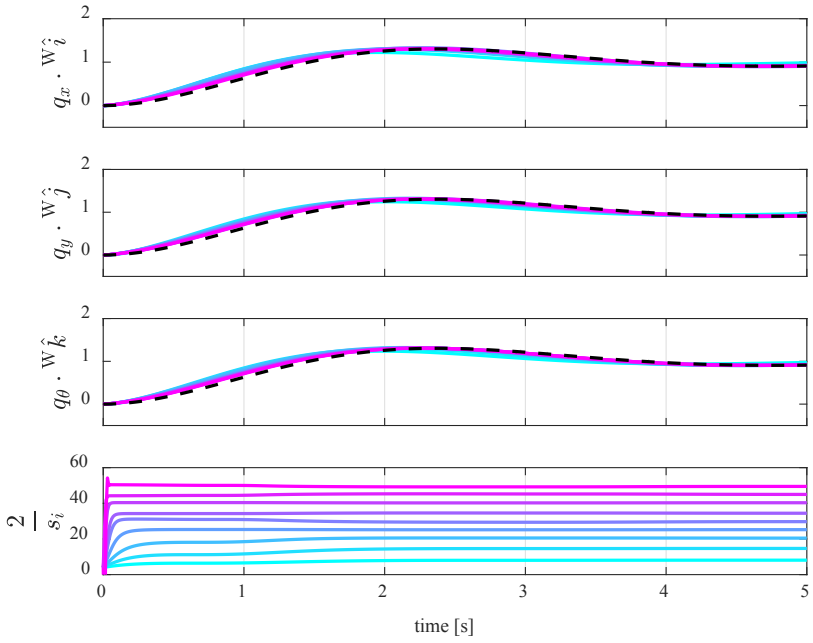

Fig. 4.4. Time responses of the object manipulated by a team of robots with different populations ranging from 10 (represented by cyan) to 50 (magenta). As depicted in the figure, an increase in the population of the team results in faster convergence of s_i.

formations (i.e. the configuration of agents around the object) to external forces along ${}^{W}\hat{i}$, ${}^{W}\hat{j}$ axes, and an external moment about ${}^{W}\hat{k}$ axis. Each line color in Fig. 4.3 represents a different team formation for a constant team population of $N = 6$. The initial response, depicted with cyan color, corresponds to a formation in which all the 6 agents are uniformly distributed in one quadrant of a circular object. The region in which the agents are distributed is gradually increased to the full circle (indicated with magenta color) with 30 degree increments. As depicted in this figure, as the formation of the group approaches to a more uniform distribution, the convergence rate of s_i increases and the system response converges to the desired impedance dynamics.

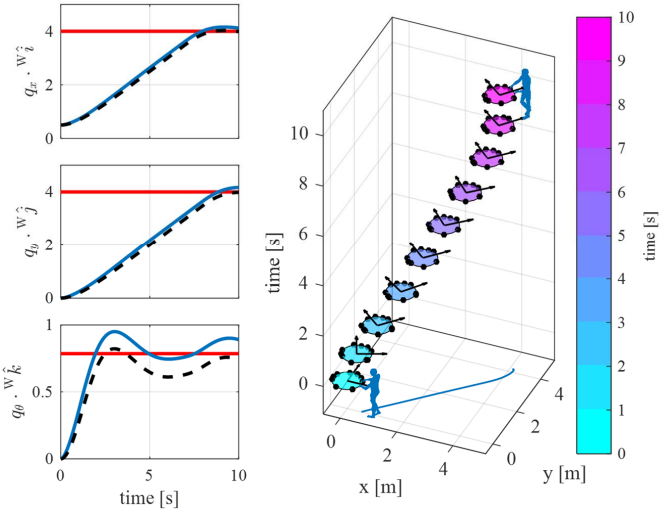

Fig. 4.5. The time response of an object with $m_o = 100$ kg, $I_o = 100$ kg.m^2 as it is carried by a human operator and a team of 10 robots. The figure on the right shows the location of the human operator and the position and orientation of the object at different instances of time. The time responses of the system and the desired impedance response (depicted with dashed black line) are illustrated on the three figure on the left hand side. The desired location of the apparent CoM of the object is illustrated with red lines on the corresponding figures.

4.3.2 Collective manipulation via impedance control

The results presented in Fig. 4.4 shows the time response of the system for a circular object with 0.5 m radius manipulated by a team of robots. Each color represents a different population of the team starting from 10 (cyan) to 50 (magenta) agents surrounding the object at random formations. The goal is to move the object to ${}^{\mathrm{W}}\bar{q}_d = [1, 1, 1]^T$ by defining ${}^{\mathrm{W}}\bar{q}_0 = {}^{\mathrm{W}}\bar{q}_d$ and ${}^{\mathrm{W}}\dot{\bar{q}}_0 = [0, 0, 0]^T$. The parameters that are used for these simulations are: ${}^{\mathrm{O}}\bar{d} = [0,\ 0,\ 0]^T$, $m_o = 10$ kg, $I_o = 10$ kg.m^2, $\mathbf{M}_d = 0.5\mathbf{I}_3$ {kg, kg.m^2}, $\mathbf{B}_d = 0.5\mathbf{I}_3$ {Ns/m, Nms/rad}, and $\mathbf{K}_d = \mathbf{I}_3$ {N/m, Nm/rad} with all the initial conditions set to zero (i.e. ${}^{\mathrm{W}}\bar{p}_0 = {}^{\mathrm{W}}\dot{\bar{p}}_0 = [0, 0, 0]^T$). As observed in Fig. 4.4, an increase in the team population results in a faster convergence of s_i. The response of the object closely follows the desired impedance response for all populations studied, while larger group populations provide better tracking.

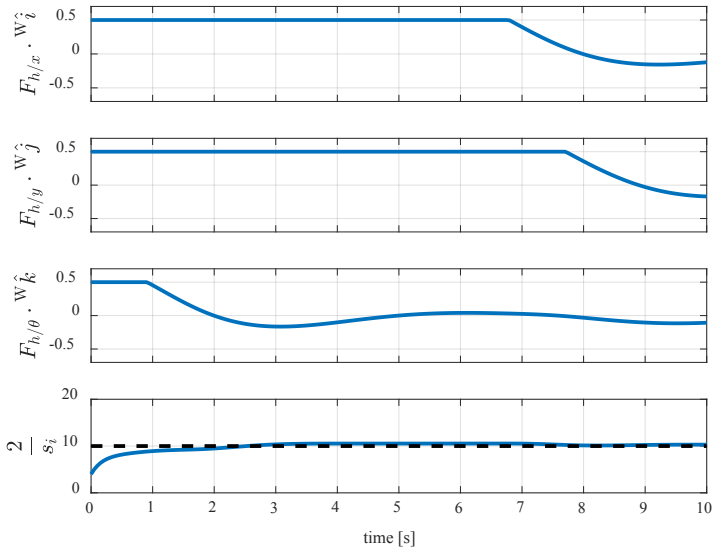

Fig. 4.6. The forces applied by the human operator and the convergence of the scaling parameter, s_i. Note $\frac{2}{s_i}$ converges to 10 which indicates a successfully estimate of the population of the team. Additionally, the human operator does not need to apply any force more than 0.5 N or any moment more than 0,5 N.m to carry a 100kg object with a mass moment of inertia of 100 kg.m^2.

4.3.3 Human-Robot coordination

One of the significant applications of collective impedance control is to provide a means for collaboration between the robotic agents and a human operator. In the scenario presented here, the human operator needs to carry a heavy object ($m_o = 100$ kg, $I_o = 100$ kg.m^2) to a desired location arbitrarily selected as (${}^{W}\bar{q} = [4, 4, \pi/4]^T$).

To perform this task, the human operator controls only their point of attachment (${}^{O}\bar{d} = [1, 0, 0]^T$) and can exert limited forces and moments to the object ($|{}^{W}\bar{F}_h| \leq 0.5$ {N, Nm}). The cooperating robotic team, which is randomly distributed around the object, needs to simplify the task for the human operator by changing the system dynamics to a desired impedance dynamics with the following parameters, selected without loss of generality: $\mathbf{M}_d = \mathbf{I}_3$ {kg, kg.m^2}, $\mathbf{B}_d = \mathbf{I}_3$ {Ns/m, Nms/rad}, and

$\mathbf{K}_d = \mathbf{0}_3$ with the apparent CoM of the object located at the point of attachment of the human operator.

The simulation results of this scenario are presented in Fig. 4.5 and Fig. 4.6. Figure 4.5 displays the time response of the system and the path of the object on the x-y plane. The response of the desired impedance dynamics is also depicted with dashed black lines. As shown in this figure, the human operator can successfully manipulate the object to the desired location with the assistance of the multi-agent system, even though the agents do not know the target position.

Figure 4.6 depicts the forces exerted to the object by the human operator and the s_i value. Note that the optimization algorithm can successfully estimate the population of the team ($N = 10$). Additionally, the human operator does not need to apply forces larger than 0.5 N or moments larger than 0.5 Nm to carry this 100 kg object.

Chapter 5

Conclusion

5.1 Discussion

This research 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 CoM of the manipulated object (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 of the interaction of the object with the environment. 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.

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 DoF 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 2.3). 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.

The proposed decetralized algorithm is also able to demonstrate impedance control capabilities which is highly required for covering interactions with the environment.

This manuscript also discusses derivation details of a collective impedance control algorithm, which does not require any information regarding the team population or formation around a known object. Additionally, a scaling parameter, s_i, is introduced to compensate for the differences between the real and assumed Jacobian of the system.

In order to eliminate the dependence of the algorithm to the population of the team (N), an estimation procedure is introduced based on the error between the desired acceleration of the apparent CoM of the object (${}^{\mathrm{W}}\ddot{\bar{q}}$) and the desired acceleration of this point based on the given impedance dynamics (${}^{\mathrm{W}}\ddot{\bar{q}}_d$) using a gradient descent algorithm. Our simulation results show that the optimization problem can successfully converge the s_i values to $2/N$ to compensate for the force scaling effects of the team population.

Further examinations of the algorithm through various simulations show that, independent of the population of the team, as the formation around the object converges to a uniform distribution, the real response of the system converges to the desired impedance response. It is also observed that the convergence rate of the s_i parameter is directly related to the population of the team.

Finally, the proposed approach is tested in a scenario, which involves the seamless coordination of a robotic team and a human operator. The objective of the robotic team is to help the human operator manipulate a heavy object to a desired location, which is only known to the operator. Our results show that the proposed algorithm successfully helps the operator carry a heavy object by replacing its dynamics with a very light object, whose apparent CoM is located at the attachment point of the operator.

Note that the algorithm presented in this manuscript is general, and can be used to substitute the response of a collectively manipulated object with any desired dynamics. Additionally, the estimation of the local scaling parameter (s_i) does not depend on the details of the desired dynamics and solely depends on ${}^{\mathrm{W}}\ddot{\bar{q}}$ and (${}^{\mathrm{W}}\ddot{\bar{q}}_d$) values. Thus, the same method can be used for different desired dynamics without modification. The results on the study of robot formation around the object suggest that the response of the system deviates from the desired response for highly asymmetric formations. Further analysis of the system in order to address this concern as well as the implementation of the proposed algorithm on physical hardware are some of the future work of this research.

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.

Bibliography

[1] Virtual coordination in collective object manipulation. `http://softrobotics.wpi.edu/tmp/WPI-SRL-2016-ColMan01.mp4`.

[2] J Scott Armstrong and Fred Collopy. Error measures for generalizing about forecasting methods: Empirical comparisons. *International journal of forecasting*, 8(1):69–80, 1992.

[3] Aaron Becker, Golnaz Habibi, Justin Werfel, Michael Rubenstein, and James 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*, pages 520–527. IEEE, 2013.

[4] Spring Berman, Quentin Lindsey, Mahmut Selman Sakar, Vijay Kumar, and Stephen C Pratt. 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):1470–1481, 2011.

[5] Paul Birkmeyer, Kevin Peterson, and Ronald S Fearing. Dash: A dynamic 16g hexapedal robot. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 2683–2689. IEEE, 2009.

[6] Michael Bonani, Valentin Longchamp, Stéphane Magnenat, Philippe Rétornaz, Daniel Burnier, Gilles Roulet, Florian Vaussard, Hannes Bleuler, and Francesco Mondada. 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*, pages 4187–4193. IEEE, 2010.

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

[8] Fabrizio Caccavale, G Giglio, Giuseppe Muscio, and Francesco Pierri. Cooperative impedance control for multiple uavs with a robotic arm. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 2366–2371. IEEE, 2015.

[9] Scott Camazine. *Self-organization in biological systems*. Princeton University Press, 2003.

[10] Alexandre Campo, Shervin Nouyan, Mauro Birattari, Roderich Groß, and Marco Dorigo. Negotiation of goal direction for cooperative transport. In *Ant Colony Optimization and Swarm Intelligence*, pages 191–202. Springer, 2006.

[11] Jianing Chen, Melvin Gauci, Wei Li, Andreas Kolling, and Roderich Gros. Occlusion-based cooperative transport with a swarm of miniature mobile robots. *Robotics, IEEE Transactions on*, 31(2):307–321, 2015.

[12] John J Craig. *Introduction to robotics: mechanics and control*, volume 3. Pearson Prentice Hall Upper Saddle River, 2005.

[13] Marco Dorigo. Swarm-bot: An experiment in swarm robotics. In *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*, pages 192–200. IEEE, 2005.

[14] Sebastian Erhart, Dominik Sieber, and Sandra Hirche. An impedance-based control architecture for multi-robot cooperative dual-arm mobile manipulation. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 315–322. IEEE, 2013.

[15] Siamak G Faal, Fuchen Chen, Weijia Tao, Mahdi Agheli, Shadi Tasdighikalat, and Cagdas D Onal. Hierarchical kinematic design of foldable hexapedal locomotion platforms. *Journal of Mechanisms and Robotics*, 8(1):011005, 2016.

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

[17] Nigel R Franks. Teams in social insects: group retrieval of prey by army ants (eciton burchelli, hymenoptera: Formicidae). *Behavioral Ecology and Sociobiology*, 18(6):425–429, 1986.

[18] Jonathan S Golan. Moore–penrose pseudoinverses. In *The Linear Algebra a Beginning Graduate Student Ought to Know*, pages 441–452. Springer, 2012.

[19] Roderich Groβ and Marco Dorigo. Evolution of solitary and group transport behaviors for autonomous robots capable of self-assembling. *Adaptive Behavior*, 16(5):285–305, 2008.

[20] Roderich Groß and Marco Dorigo. Group transport of an object to a target that only some group members may sense. In *Parallel Problem Solving from Nature-PPSN VIII*, pages 852–861. Springer, 2004.

[21] Golnaz Habibi, Lauren Schmidt, Mathew Jellins, and James McLurkin. K-redundant trees for safe and efficient multi-robot recovery in complex environments. In *Robotics Research*, pages 149–165. Springer, 2016.

[22] Moussa Haddad, Wisama Khalil, and HE Lehtihet. Trajectory planning of unicycle mobile robots with a trapezoidal-velocity constraint. *Robotics, IEEE Transactions on*, 26(5):954–962, 2010.

[23] Neville Hogan. Impedance control: An approach to manipulation.

[24] Neville Hogan. Impedance control: An approach to manipulation: Part iiimplementation. *Journal of dynamic systems, measurement, and control*, 107(1):8–16, 1985.

[25] Rob J Hyndman and Anne B Koehler. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688, 2006.

[26] Shadi T Kalat, Siamak G. Kalat Faal, and Cagdas D Onal. Scalable collective impedance control of an object via a decentralized force control method. In *The 2017 American Control Conference*. ACC, 2017.

[27] Shadi Tasdighi Kalat, Siamak G Faal, Ugur Celik, and Cagdas D Onal. Tribot: A minimally-actuated accessible holonomic hexapedal locomotion platform. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 6292–6297. IEEE, 2015.

[28] Yuichi Kobayashi and Shigeyuki Hosoe. Cooperative enclosing and grasping of an object by decentralized mobile robots using local observation. *International Journal of Social Robotics*, 4(1):19–32, 2012.

[29] Michael JB Krieger, Jean-Bernard Billeter, and Laurent Keller. Ant-like task allocation and recruitment in cooperative robots. *Nature*, 406(6799):992–995, 2000.

[30] C Ronald Kube and Eric Bonabeau. Cooperative transport by ants and robots. *Robotics and autonomous systems*, 30(1):85–101, 2000.

[31] C Ronald Kube and Hong Zhang. Collective robotics: From social insects to robots. *Adaptive behavior*, 2(2):189–218, 1993.

[32] David W Meer. *Experiments in cooperative manipulation of flexible objects*. PhD thesis, stanford university, 1994.

[33] S Ali A Moosavian and Evangelos Papadopoulos. Multiple impedance control for object manipulation. In *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, volume 1, pages 461–466. IEEE, 1998.

[34] PSG de M NETO, Ricardo de A Araújo, Gustavo G Petry, Tiago AE Ferreira, and Germano C Vasconcelos. Hybrid swarm system for time series forecasting. *VI Encontro Nacional de Inteligência Artificial (ENIA)*, 2007.

[35] Stephan Olariu and Albert Y Zomaya. *Handbook of bioinspired algorithms and applications*. CRC Press, 2005.

[36] Guilherme AS Pereira, Mario FM Campos, and Vijay Kumar. Decentralized algorithms for multi-robot manipulation via caging. *The International Journal of Robotics Research*, 23(7-8):783–795, 2004.

[37] Guilherme AS Pereira, Vijay Kumar, John R Spletzer, Camillo J Taylor, and Mario FM Campos. Cooperative transport of planar objects by multiple mobile

robots using object closure. In *Experimental Robotics VIII*, pages 287–296. Springer, 2003.

[38] Michael Rubenstein, Christian Ahler, and Radhika Nagpal. Kilobot: A low cost scalable robot system for collective behaviors. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3293–3298. IEEE, 2012.

[39] Michael Rubenstein, Adrian Cabrera, Justin Werfel, Golnaz Habibi, James McLurkin, and Radhika 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*, pages 47–54. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[40] Erol Şahin. Swarm robotics: From sources of inspiration to domains of application. In *Swarm robotics*, pages 10–20. Springer, 2004.

[41] Stanley A Schneider and Robert H Cannon. Object impedance control for cooperative manipulation: Theory and experimental results. *IEEE Transactions on Robotics and Automation*, 8(3):383–394, 1992.

[42] Peng Song and Vijay Kumar. A potential field based approach to multi-robot manipulation. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 2, pages 1217–1222. IEEE, 2002.

[43] J Spletzer, Aveek K Das, Rafael Fierro, Camillo J Taylor, Vijay Kumar, and James P Ostrowski. Cooperative localization and control for multi-robot manipulation. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 2, pages 631–636. IEEE, 2001.

[44] Attawith Sudsang and Jean Ponce. A new approach to motion planning for disc-shaped robots manipulating a polygonal object in the plane. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 2, pages 1068–1075. IEEE, 2000.

[45] Attawith Sudsang, Fred Rothganger, and Jean Ponce. Motion planning for disc-shaped robots pushing a polygonal object in the plane. *Robotics and Automation, IEEE Transactions on*, 18(4):550–562, 2002.

[46] Efstathios Velenis and Panagiotis Tsiotras. Optimal velocity profile generation for given acceleration limits: theoretical analysis. *system*, 2:5, 2005.

[47] Weiwei Wan, Rui Fukui, Masamichi Shimosaka, Tomomasa Sato, and Yasuo Kuniyoshi. Cooperative manipulation with least number of robots via robust caging. In *Advanced Intelligent Mechatronics (AIM), 2012 IEEE/ASME International Conference on*, pages 896–903. IEEE, 2012.

[48] ZhiDong Wang, Yasuhisa Hirata, and Kazuhiro Kosuge. Control multiple mobile robots for object caging and manipulation. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 2, pages 1751–1756. IEEE, 2003.

[49] ZhiDong Wang and Vijay Kumar. Object closure and manipulation by multiple cooperating mobile robots. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 1, pages 394–399. IEEE, 2002.

[50] Zijian Wang and Mac Schwager. Multi-robot manipulation without communication. In *International Symposium on Distributed Autonomous Robotic Systems (DARS)*, 2014.

[51] Zijian Wang and Mac Schwager. Kinematic multi-robot manipulation with no communication using force feedback. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[52] Cort J Willmott. On the validation of models. *Physical geography*, 2(2):184–194, 1981.

[53] Michael M Zavlanos, Ali Jadbabaie, and George J Pappas. Flocking while preserving network connectivity. In *Decision and Control, 2007 46th IEEE Conference on*, pages 2919–2924. IEEE, 2007.