Decentralized Control of Collective Transport by Multi-Robot Systems with Minimal Information

by

Hamed Farivarnejad

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved October 2020 by the
Graduate Supervisory Committee:

Spring Berman, Chair
Marc Mignolet
Konstantinos Tsakalis
Panagiotis Artemiadis
Stephanie Gil

ARIZONA STATE UNIVERSITY

December 2020

ABSTRACT

One potential application of multi-robot systems is collective transport, a task in which multiple mobile robots collaboratively transport a payload that is too large or heavy to be carried by a single robot. Numerous control schemes have been proposed for collective transport in environments where robots can localize themselves (e.g., using GPS) and communicate with one another, have information about the payload's geometric and dynamical properties, and follow predefined robot and/or payload trajectories. However, these approaches cannot be applied in uncertain environments where robots do not have reliable communication and GPS and lack information about the payload. These conditions characterize a variety of applications, including construction, mining, assembly in space and underwater, search-and-rescue, and disaster response.

Toward this end, this thesis presents decentralized control strategies for collective transport by robots that regulate their actions using only their local sensor measurements and minimal prior information. These strategies can be implemented on robots that have limited or absent localization capabilities, do not explicitly exchange information, and are not assigned predefined trajectories. The controllers are developed for collective transport over planar surfaces, but can be extended to three-dimensional environments.

This thesis addresses the above problem for two control objectives. First, decentralized controllers are proposed for velocity control of collective transport, in which the robots must transport a payload at a constant velocity through an unbounded domain that may contain strictly convex obstacles. The robots are provided only with the target transport velocity, and they do not have global localization or prior information about any obstacles in the environment. Second, decentralized controllers are proposed for position control of collective transport, in which the robots must transport a payload to a target position through a bounded or unbounded domain that may contain convex obstacles. The robots are subject to the same constraints as in the velocity control scenario, except that they are assumed to have global localization. Theoretical guarantees for successful execution of the task are derived using techniques from nonlinear control theory, and it is shown through simulations and physical robot experiments that the transport objectives are achieved with the proposed controllers.

DEDICATION

To my dear family

ACKNOWLEDGEMENTS

I would first like to thank my advisor Dr. Spring Berman for all her kind help and support during my entire PhD. She has been a constant source of encouragement throughout my student years. I am especially grateful to her for allowing me the flexibility to take this research in a direction of my choosing and for motivating me to keep going on this path.

Secondly, I would like to thank Dr. Konstantinos Tsakalis for teaching nonlinear control that I used the most to do the analysis in this work. I would also like to thank Dr. Panagiotis Artemiadis, for teaching a useful class on digital control, and being on my committee.

A heartful thanks to Dr. Marc Mignolet for kindly answering my questions in dynamics, being in my committee, and for helping me along critical points of my graduate studies. I would like also to acknowledge Dr. Stephanie Gil for accepting to be in my committee.

I would like to thank my friend Amir for his immense help with running the simulations and the experiments of this work, and proofreading of this thesis. I am also thankful to my friends, Azadeh, Shiba, Karthik, Aniket, Zahi, Sean, Ruben, Ragesh, and Rakshith, for making these years enjoyable. I will forever remember our Friday evening beer sessions.

Finally, I would like to acknowledge my parent's and sister's unconditional love and support.

This research was supported by DARPA Young Faculty Award D14AP00054, Office of Naval Research (ONR) Young Investigator Award N00014-16-1-2605, and the Arizona State University Global Security Initiative (GSI).

TABLE OF CONTENTS

Page

APPENDIX

LIST OF FIGURES

Figure Page

Figure Page

Chapter 1

INTRODUCTION

Collective transport is a potential application of multi-robot systems. When a payload is too large or heavy to be carried by a single robot, a team of robots can be deployed to collaboratively transport the payload to a desired destination, as shown in Fig. 1.1. This type of task can arise in a variety of applications, including construction and manufacturing, assembly in space and underwater, search-and-rescue operations, and disaster response. There has been a significant amount of research on collective transport in the past three decades, and a wide range of controllers have been proposed for control and coordination of a group of robots performing collective transport tasks. The proposed control schemes often rely on assumptions that restrict the autonomy of the robots that execute the task. For instance, many of the proposed controllers require the robots to have information about the payload's geometry and dynamics, as well as measurements of its position and velocity. There are numerous control methods that rely on explicit communication between the robots or predefined trajectories for the payload and/or the robots. To this end, this thesis focuses on enabling autonomous execution of collective transport by robots that lack the capabilities and information required by these prior approaches. In this chapter, we first highlight the main contributions of the thesis in Section 1.1. We then review the literature on collective transport in Section 1.2 and relevant literature on obstacle avoidance in Section 1.3. We conclude this chapter with the organization of the thesis

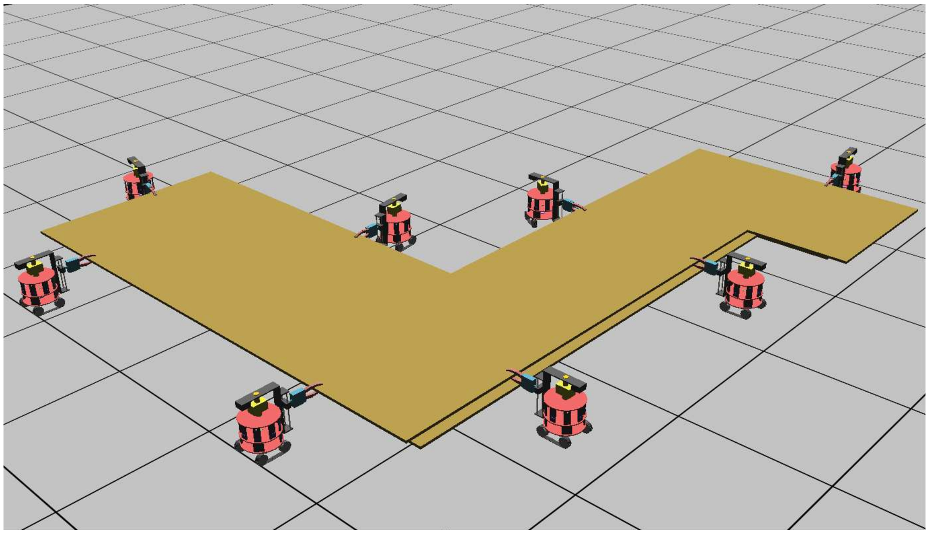


Figure 1.1: A Simulation of a Collective Transport Task by *Pheeno* Robots (Wilson *et al.*, 2016) in the 3D Robot Simulator Webots (Michel, 2004).

in Section 1.4.

1.1 Contributions

We propose completely decentralized robot controllers that can achieve collective transport with a quantifiable degree of predictability in unknown, remote, and hazardous environments with limited data and communication. Our approach is inspired by group food retrieval in ants (Czaczkes and Ratnieks, 2013; McCreery and Breed, 2014; Gelblum *et al.*, 2016). This behavior is a striking example of decentralized collective transport by autonomous individuals, since the transport team members do not follow predefined paths, use explicit communication, or have prior knowledge about the payload, the distribution of teammates around it, or the locations and shapes of obstacles in the environment (Medina *et al.*, 2020). While the ants know the direction to their nest, it is likely that their activities during transport are influenced only by their local information. Analogously, the control strategies that we develop must rely on minimal information and exhibit robustness to uncertainties in the payload dynamics and to external disturbances. The proposed controllers for collective transport in this thesis have the following characteristics:

- The robots have identical capabilities and information.
- The robots do not communicate with one another.
- There are no predefined trajectories for the robots and/or the payload.
- The robots do not have information about the payload's geometry or dynamics, and they do not have feedback on its position and velocity over time.
- The robots do not have information about the size of the transport team and the distribution of robots around the load.
- The robots do not have information about the positions and shapes of any obstacles in the environment.
- The proposed controllers are completely decentralized, in that each robot uses only its local measurements to calculate the control law.
- The controllers have theoretical guarantees on stability, convergence properties, and collision avoidance.

Two scenarios are studied in this thesis. In the first scenario, the objective is to stabilize the translational velocity of the payload to a constant value and its angular velocity to zero. The transport is assumed to be done in an unbounded domain that may contain strictly convex obstacles. Each robot can measure only its own velocity and its distance from any obstacles in its sensing range. The proposed controller is proved to safely steer the payload around obstacles and eventually drive its velocity to the target value. In the second scenario, the robots must stabilize the payload's center of mass to a target position and stabilize its angular velocity to zero. The domain may be unbounded without obstacles, or bounded and convex with convex obstacles. Each robot can measure its own position in a global coordinate frame, in addition to its velocity and its distance from any nearby obstacles. It is shown that with the proposed controllers, the payload is safely transported around obstacles toward the target position and converges to a neighborhood of this position.

1.2 Review of Literature on Collective Transport

Over the past three decades, various control strategies have been proposed for collective transport in scenarios that are not subject to all of the constraints listed in the previous subsection. In some centralized control approaches, such as (Shahrokhi and Becker, 2016), a supervisor (a human or a central computer) observes the motion of the system and communicates appropriate control commands to the robots in order to guide the payload toward the goal. Decentralized control strategies, which can improve the system's robustness to errors, failures, and disturbances, have been investigated more extensively. Some proposed methods are leader-follower algorithms in which the leader takes the main role in planning the payload trajectory and controlling its motion to the goal, while the other robots contribute to the transport motion in a coordinated manner. For instance, in (Wang and Schwager, 2014) and (Wang and Schwager, 2015), a consensus-based approach is presented in which the leader is more powerful than the followers and is provided with a predefined path to the goal, and the followers, which do not know anything about the leader's intention, can effectively attain a consensus on the magnitude and the direction of the force they have to apply to the load. Also, in (Wang and Schwager, 2016), a leader robot applies a force to move the load over a predefined path, and followers can estimate the direction of the object movement using force sensing at the attachment point and apply their forces along this estimated direction in order to assist the leader. Considering the load as the leader is another approach presented in (Yufka and Ozkan, 2015), where the followers (the transporting robots) use a path planning approach to preserve their initial position and orientation with respect to the virtual leader (load) during the transport.

In many decentralized control approaches to collective transport, all robots in the transport team are assumed to be identical. In (Chen *et al.*, 2015), a decentralized approach is proposed for cooperative transport when the load is significantly larger than the robots, and the robots push the load only if their line of sight to the goal is occluded by the object. In (Habibi *et al.*, 2015), four algorithms are presented that enable the robots to estimate the centroid of the load, rotate the load, and transport it over certain marked points that can be recognized by a guide robot. Transporting a flexible payload is considered in (Bai and Wen, 2009) and (Bai and Wen, 2010), where the reaction force between the robot and the payload is modeled as the gradient of a nonlinear potential that describes the load deformation. Group food retrieval in ants is studied in (Berman *et al.*, 2010), in which a vision-based force sensor is used to collect data on ants during transport and a hybrid dynamical model is developed to replicate the observed collective behavior. In (Bais *et al.*, 2015), the load weight is distributed among robots with heterogeneous load-carrying capabilities, and the load is driven along a desired trajectory. A decentralized distribution of the forces applied to the load is proposed in (Kalat *et al.*, 2018), where each robot relies on the behavior of a cooperative virtual teammate. In (Rubenstein *et al.*, 2013), it is assumed that all the robots know the target direction to the goal, and a simple control law, which uses just the robot's velocity, is developed to calculate the force that the robot has to apply to the load. In a similar scenario, decentralized PID controllers were used in (Wilson *et al.*, 2016) for collective transport by three small mobile robots. In (Culbertson and Schwager, 2018), a decentralized approach is proposed for cooperative manipulation in which the robots have a common reference model for the desired payload motion and use an adaptive controller to compensate for the effect of friction on the payload. Whereas this approach requires the robots to have access to measurements of the payload's linear and angular velocities, ours does not require any information on the payload's motion.

Some methods require robots to communicate their measurements to each other in order to estimate parameters of an unknown payload (Franchi *et al.*, 2014; Marino *et al.*, 2017). More recently, (Dohmann and Hirche, 2020) proposes an event-triggered communication strategy with distributed impedance control to improve the stability and robustness of cooperative manipulation of unknown payloads in unknown environments. Other approaches do not rely on communication or prior information about the payload's dynamics (Marino and Pierri, 2018), but they require a supervisor to define trajectories beforehand for the robots and the payload (Tsiamis *et al.*, 2015; Gueaieb *et al.*, 2003; Li *et al.*, 2015). Recently, adaptive robust control approaches have been proposed for planar and three-dimensional manipulation (Kim *et al.*, 2018; Lee *et al.*, 2018; Sadati and Ghaffarkhah, 2007; Dai and Liu, 2017; Lee *et al.*, 2017; Marino, 2017; Pliego-Jimenez and Arteaga-Perez, 2017). These approaches combine a stabilizing term with a regression term in the controller in order to achieve stabilization in the presence of parameter uncertainties. However, the approaches require either prior information about the robots' distribution around the payload or feedback on the payload's motion. In more recent work (Jin *et al.*, 2018), robots jointly reach the same desired motion by running a time-varying quadratic program which is solved online by a neural network scheme.

Recently, learning schemes have also been proposed for cooperative manipulation. In (Jin *et al.*, 2018), robots in a transport team, which explicitly exchange information, jointly reach the same desired motion by running a time-varying quadratic program which is solved online by a neural network scheme. A dynamic recurrent neural network is used in (Li *et al.*, 2020) to solve a quadratic program, which computes cooperative kinematic controllers for redundant manipulators using partially known information about the payload and the teammates. In addition, reinforcement learning is used in (Ding *et al.*, 2020) to design two distributed approaches to cooperative manipulation: the first applies Q-learning with individual reward functions, and the second utilizes game-theoretic techniques. The first approach exhibits more robustness to different reward structures than the second.

1.3 Review of Literature on Obstacle Avoidance

Obstacle avoidance has been a challenging topic in the control of robotic systems and has been extensively studied by researchers over the past few decades. Numerous approaches have been proposed to prevent robots from colliding with obstacles in their workspace, ranging from heuristic solutions (Xi *et al.*, 2005) to algorithmically rigorous (LaValle, 2006; Lindemann and LaValle, 2009; Karaman and Frazzoli, 2011; Belta *et al.*, 2005) and/or mathematically rigorous (Connolly *et al.*, 1990; Kim and Khosla, 1992; Rimon and Koditschek, 1992) motion planning and control schemes. These approaches can be categorized according to their requirements on the robot's localization capabilities and prior knowledge about the environment. We first describe key developments in obstacle-avoidance control schemes, along with their requirements, and then summarize our contribution in the context of this prior work.

Many existing obstacle avoidance strategies require the robot to have global localization as well as information about the exact shapes and locations of the obstacles. One pioneering solution with these requirements, first proposed in (Khatib, 1986) in the 1980's, uses the concept of virtual potential fields. Subsequent approaches based on potential fields include (Shahidi *et al.*, 1991), which assumes an environment that contains circular obstacles with known centers and radii, and (Kim and Khosla, 1992), in which harmonic potential fields, which satisfy Laplace's equation, are used to guarantee collision-free robot navigation to a target position on the domain boundary. The construction of potential fields called *navigation functions* on bounded manifolds was a significant development that enabled the design of control laws for exact robot navigation to destinations in generalized sphere worlds (Rimon and Koditschek, 1992; Koditschek and Rimon, 1990). These control laws require accurate robot localization and prior information about the locations of the obstacles and the equations of their boundaries. Numerous works have adapted the navigation function approach to different scenarios. In (Conner *et al.*, 2003), a combination of harmonic potentials and navigation functions is proposed as a solution when the free space can be decomposed into a chain of connected polygons. In (Ogren and Leonard, 2005), a navigation function-based strategy is merged with the dynamic window approach (Fox *et al.*, 1997) to produce faster robot convergence to a destination in dynamic environments. An algorithm for automatic tuning of the parameters of navigation functions for sphere worlds is presented in (Filippidis and Kyriakopoulos, 2011). In (Li and Tanner, 2019), navigation functions are designed such that the robot can asymptotically track a moving target in environments with obstacles. Recently, a modified navigation function-based approach was proposed in (Paternain *et al.*, 2018) to produce robot convergence to the minimum of a globally convex potential function in an environment with arbitrary convex obstacles.

Control schemes that use *barrier certificates* (Prajna *et al.*, 2004) and *barrier functions* (Ames *et al.*, 2017) have been recently developed for scenarios where there are unsafe or undesired regions in a dynamical system's state space that its trajectories must avoid. These methods require knowledge of the exact boundary of the unsafe or undesired region, which is the set of obstacles when the objective is collision-free robot navigation. In (Wang *et al.*, 2017), a control barrier function scheme is proposed to prevent collisions among the robots in a swarm, and it also prevents collisions between the robots and static or dynamic obstacles. This control approach requires knowledge of the centers and radii of the circles that virtually bound the obstacles.

Another category of work on obstacle avoidance can be characterized by the dependence of the proposed control strategies on only approximate knowledge about the locations and geometries of the obstacles. In (Guldner and Utkin, 1995), a sliding mode controller is presented for tracking the gradient of potential fields that are constructed based on the smallest circle that bounds each obstacle. In the recent work (Paternain and Ribeiro, 2017), a stochastic navigation function-based approach is proposed that requires *a priori* estimates of the obstacle geometries and locations according to a probability distribution (a belief space). It is shown that if the robot follows a stochastic approximation of the gradient of the navigation function, convergence to the desired destination and obstacle avoidance are guaranteed with a probability of one. The recent work (Arslan and Koditschek, 2019) proposes a sensor-based feedback law that uses a Voronoi diagram for the environment which the robot computes online. While this approach applies to environments with unknown convex obstacles, it requires an assumption on the obstacle curvature (*Assumption 2* in (Arslan and Koditschek, 2019)).

Other obstacle avoidance strategies do not require prior knowledge about the obstacles, but are subject to different restrictions or rely on other available information. A modified potential field-based method is presented in (Ge and Cui, 2000) for the case where the target robot position is very close to one of the obstacles, and it is extended to environments with moving obstacles in (Ge and Cui, 2002). Even though the proposed controller does not require any knowledge about the shapes and positions of the obstacles, it cannot eliminate all local minima in the environment that can trap the robot. In (Ramírez-Llanos and Martínez, 2019), a stochastic source-seeking scheme is proposed for a GPS-denied environment with a signal that is directly measurable by the robot. The robot is allowed to contact the boundaries of the environment and the obstacles and travels around these boundaries, maintaining contact with them, until it finds a feasible direction to the source of the signal in the free space.

In addition to the works described above, which focus on designing controllers with theoretical guarantees in particular types of environments, numerous other works focus on developing obstacle avoidance strategies that, while not necessarily amenable to theoretical analysis, are convenient to implement using typical sensors on physical robotic platforms. For example, visual sensing approaches for estimating the distance and velocity of nearby obstacles are described in (Rassameepaiboon and Assawinchaichote, 2018) and (Padhy *et al.*, 2019) for terrestrial and aerial applications, respectively. The work (Ferro *et al.*, 2019) proposes a combination of a visual servoing control scheme and a velocity estimation algorithm for obstacle avoidance by a legged robot and an omnidirectional wheeled robot.

1.4 Organization of the Thesis

This thesis is organized in two parts. The first part is about the velocity control of collective transport in an unbounded domain. The second part focuses on position control of collective transport in a bounded convex domain.

The first part covers Chapter 2 and Chapter 3. Chapter 2 describes controllers for decentralized velocity control of collective transport in an unbounded domain without obstacles. This chapter includes the results from (**Farivarnejad, H.** *et al.*, 2016; **Farivarnejad, H.** and Berman, 2018, 2020a). Chapter 3 proposes an obstacle avoidance controller for a single robot that must converge to a desired velocity in an unbounded domain. This chapter includes the results from (**Farivarnejad, H.** and Berman, 2020c). Combining the results from Chapter 2 and Chapter 3, a decentralized control law is proposed for safe velocity control of collective transport in unbounded domains that contain strictly convex obstacles.

The second part has the same organization as the first part. Chapter 4 proposes a decentralized controller for position control of collective transport in an unbounded domain without obstacles. This chapter includes the results from (**Farivarnejad, H.** and Berman, 2020b). Chapter 5 first introduces a novel method for obstacle avoidance by a single robot in a bounded convex domain. This chapter then combines the decentralized controller proposed in Chapter 4 with the proposed obstacle avoidance controller to introduce a decentralized controller for safe position control of collective transport in a bounded convex domain that contains convex obstacles. Chapter 5 includes the results from (**Farivarnejad, H.** *et al.*, 2020) and (**Farivarnejad, H.** and Berman, 2021).

Finally, Chapter 6 concludes with future work and possible extensions to the research presented in this thesis.

Chapter 2

VELOCITY CONTROL OF COLLECTIVE TRANSPORT IN UNBOUNDED DOMAINS WITHOUT OBSTACLES

In this chapter, we propose decentralized controllers for velocity regulation of collective transport in unbounded environments without obstacles. The objective is for the robots to transport a payload at a constant linear velocity and zero angular velocity. This control strategy could, for example, be applied to tasks in which the robots must convey a payload along a sequence of straight paths, each associated with a target transport speed and heading.

We propose three decentralized control schemes. The first is a sliding mode controller that is initially designed for point-mass robots and then is modified for nonholonomic robots. The second controller is a proportional-integral control law that is designed for velocity stabilization of point-mass robots and elimination of payload drift from a desired path. The third controller is an adaptive control scheme for nonholonomic robots that grasp the payload with a single-DOF manipulator.

Note: Dr. Sean Wilson conducted the experiments in Section 2.1.6.

2.1 Decentralized Sliding Mode Control

In this section, we present, analyze, and implement a decentralized control scheme based on a sliding mode control approach for scenarios that are similar to those in (Wilson *et al.*, 2016) and (Rubenstein *et al.*, 2013). We design controllers for the robots in the transport team and prove that the payload converges to the target velocity under the action of these controllers. Although sliding mode control has previously been used for cooperative manipulation in (Yagiz *et al.*, 2010; Yim *et al.*, 1999; Parra-Vega *et al.*, 2013; Ponce-Hinestroza *et al.*, 2016), these strategies require predefined trajectories for each robot and/or for the payload. In contrast, the control strategy proposed here only requires a predefined target transport velocity and local robot measurements of their own velocity and heading, and it does not rely on information about the environment, load, or transport team.

2.1.1 Problem Statement

We consider a team of identical autonomous ground robots, each equipped with a manipulator arm, that are arranged on a planar surface in an arbitrary configuration around a payload. The robots are all grasping the load and holding it above the ground (as in Fig. 1.1). We assume that each robot can measure its own speed and heading. The robots do not have global localization or communication capabilities, and they lack information about the payload dynamics, the number of robots in the transport team and their distribution around the payload, and the layout of the environment.

Our objective is to design decentralized controllers that will drive the team of robots to collectively transport the load at a desired speed along a straight path in a target direction, with zero angular velocity. The target direction is defined as the direction of a specified line that passes through the payload's center of mass, illustrated by the dashed green line in Fig. 2.1. We assume that each robot knows the target speed and target direction, although they are not assigned predefined trajectories. (We note that the controllers will drive each robot to move along the line that passes through its initial position and is parallel to the target direction.) The robots do not require information about the position of the payload's center of mass or their positions relative to the center of mass. To enable the robots to act autonomously during transport, we do not assign them reference speed profiles that would require the presence of a global supervisor with knowledge about their positions with respect to the goal and their distribution around the payload. Instead, the controllers must depend only on the minimal information that is available to the robots and should be robust to the uncertainties in the highly nonlinear dynamics of the manipulated payload.

2.1.2 Dynamical Model

We consider a load that is transported in the plane by a group of N robots, each of which is modeled as a point-mass agent. The position of robot i at time t in an inertial reference frame is given by $\boldsymbol{x}_i(t) \in \mathbb{R}^2$. The robot's actuating force is denoted by $\boldsymbol{u}_i \in \mathbb{R}^2$, and the reaction force exerted by the load on the robot is $\boldsymbol{F}_i \in \mathbb{R}^2$. Given that robot i has mass m_i, the dynamics of the robot are:

$$m_i \ddot{\boldsymbol{x}}_i = \boldsymbol{u}_i - \boldsymbol{F}_i \ . \tag{2.1}$$

In order to develop a sliding mode controller for robot i, we must be able to write the robot's dynamics in the form

$$\ddot{\boldsymbol{x}}_i = \boldsymbol{h} + \boldsymbol{G}\boldsymbol{u}_i \ , \tag{2.2}$$

in which $\boldsymbol{G}$ is an input matrix that is a function of the load dynamics, and $\boldsymbol{h}$ is a nonlinear term that describes the effects of both the load dynamics and the forces applied by the other robots. The sliding mode controller will only require bounds on this nonlinear term, not a precise characterization. In the remainder of this section, we show that Eq. 2.1 can be put into the form Eq. 2.2.

The notation for our dynamical model of collective transport is shown in Fig. 2.1. We define an inertial coordinate frame I and a local coordinate frame B that is fixed to the load. The matrix $\boldsymbol{R}_B^I$ is the rotation matrix from coordinate frame B to

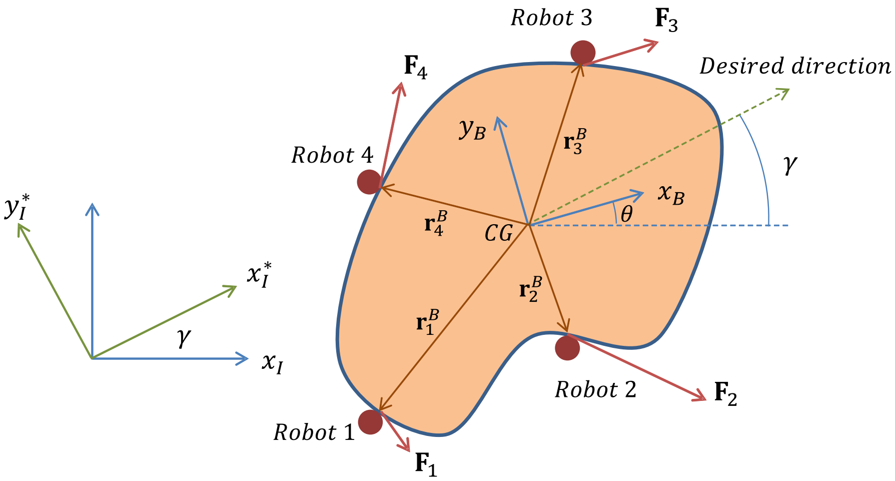


Figure 2.1: Schematic Representation of a Collective Transport Task with Four Robots and the Associated Coordinate Systems.

coordinate frame I. We define $\ddot{\boldsymbol{x}}_o^B$ and $\boldsymbol{r}_i^B$, both expressed in coordinate frame B, as the acceleration of the load's center of gravity (CG) and the vector from the load's CG to the attachment point of robot i, respectively. We denote the load's orientation in frame I by θ_o, its angular velocity by ω_o, and its angular acceleration by α_o. We now recall that the cross product of any two non-zero vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ can be expressed as $\boldsymbol{a} \times \boldsymbol{b} = \hat{\boldsymbol{a}}\boldsymbol{b}$, where $\hat{\boldsymbol{a}}$ is a skew-symmetric matrix. Using this notation, we denote the skew-symmetric matrix representations of ω_o and α_o by $\hat{\omega}_o$ and $\hat{\alpha}_o$, respectively.

Since we assume that each robot is rigidly attached to the load, the acceleration of robot i can be written in terms of the load's angular velocity and angular acceleration as follows:

$$\ddot{\boldsymbol{x}}_i = \boldsymbol{R}_B^I(\ddot{\boldsymbol{x}}_o^B + \hat{\alpha}_o\boldsymbol{r}_i^B + \hat{\omega}_o(\hat{\omega}_o\boldsymbol{r}_i^B)) \ . \tag{2.3}$$

Noting that

$$\hat{\alpha}_o\boldsymbol{r}_i^B = -\hat{\boldsymbol{r}}_i^B\alpha_o = (\hat{\boldsymbol{r}}_i^B)^T\alpha_o \tag{2.4}$$

and that $\alpha_o = \ddot{\theta}_o$, we can rewrite Eq. 2.3 as:

$$\ddot{\boldsymbol{x}}_i = \boldsymbol{R}_B^I\left(\begin{bmatrix}\boldsymbol{I} & (\hat{\boldsymbol{r}}_i^B)^T\end{bmatrix}\begin{bmatrix}\ddot{\boldsymbol{x}}_o^B \\ \ddot{\theta}_o\end{bmatrix} + \hat{\omega}_o(\hat{\omega}_o\boldsymbol{r}_i^B)\right) \ , \tag{2.5}$$

where $\boldsymbol{I}$ is the identity matrix. We now represent the load's translational and rotational dynamics together in the following matrix form:

$$\begin{aligned}\begin{bmatrix} m_o\boldsymbol{I} & 0 \\ 0 & I_o \end{bmatrix}\begin{bmatrix} \ddot{\boldsymbol{x}}_o^B \\ \ddot{\theta}_o \end{bmatrix} &= \begin{bmatrix} \boldsymbol{I} & \cdots & \boldsymbol{I} \\ \hat{\boldsymbol{r}}_1^B & \cdots & \hat{\boldsymbol{r}}_N^B \end{bmatrix}\boldsymbol{\mathcal{F}}^B, \\ \boldsymbol{\mathcal{F}}^B &= \begin{bmatrix} (\boldsymbol{F}_1^B)^T & \cdots & (\boldsymbol{F}_N^B)^T \end{bmatrix}^T, \end{aligned} \tag{2.6}$$

where $\boldsymbol{F}_i^B$ is force $\boldsymbol{F}_i$ in coordinate frame B, m_o is the mass of the load, and I_o is the load's moment of inertia about the axis normal to the plane of the motion and passing through its CG. Solving for the load's acceleration vector from this equation and substituting it into Eq. 2.5, we obtain the acceleration of robot i as:

$$\ddot{\boldsymbol{x}}_i \;=\; \boldsymbol{R}_B^I \begin{bmatrix} \boldsymbol{I} & (\hat{\boldsymbol{r}}_i^B)^T \end{bmatrix} \boldsymbol{M}_o^{-1} \begin{bmatrix} \boldsymbol{I} & \cdots & \boldsymbol{I} \\ \hat{\boldsymbol{r}}_1^B & \cdots & \hat{\boldsymbol{r}}_N^B \end{bmatrix} \boldsymbol{\mathcal{F}}^B + \; \boldsymbol{R}_B^I \hat{\omega}_o (\hat{\omega}_o \boldsymbol{r}_i^B) \;, \tag{2.7}$$

where

$$\boldsymbol{M}_o = \begin{bmatrix} m_o\boldsymbol{I} & 0 \\ 0 & I_o \end{bmatrix}. \tag{2.8}$$

We can rewrite Eq. 2.7 as the sum of two terms, one containing the force applied by robot i and the other containing the forces applied by all the other robots:

$$\ddot{\boldsymbol{x}}_i = \boldsymbol{Q} + \boldsymbol{P}\boldsymbol{F}_i \;, \tag{2.9}$$

where

$$\boldsymbol{P} = \boldsymbol{R}_B^I \begin{bmatrix} \boldsymbol{I} & (\hat{\boldsymbol{r}}_i^B)^T \end{bmatrix} \boldsymbol{M}_o^{-1} \begin{bmatrix} \boldsymbol{I} \\ \hat{\boldsymbol{r}}_i^B \end{bmatrix} \boldsymbol{R}_I^B \tag{2.10}$$

and

$$\boldsymbol{Q} = \boldsymbol{R}_B^I \boldsymbol{Q}_1 \boldsymbol{Q}_2 \boldsymbol{\mathcal{F}}_r^B + \boldsymbol{R}_B^I \hat{\omega}_o (\hat{\omega}_o \boldsymbol{r}_i^B) \;, \tag{2.11}$$

$$\boldsymbol{Q}_1 = \begin{bmatrix} \boldsymbol{I} & (\hat{\boldsymbol{r}}_i^B)^T \end{bmatrix} \boldsymbol{M}_o{}^{-1} \ ,$$

$$\boldsymbol{Q}_2 = \begin{bmatrix} \boldsymbol{I} & \cdots & \cdots & \cdots & \dots & \boldsymbol{I} \\ \hat{\boldsymbol{r}}_1^B & \cdots & \hat{\boldsymbol{r}}_{i-1}^B & \hat{\boldsymbol{r}}_{i+1}^B & \cdots & \hat{\boldsymbol{r}}_N^B \end{bmatrix} \ ,$$

$$\boldsymbol{\mathcal{F}}_r^B = \begin{bmatrix} (\boldsymbol{F}_1^B)^T & \cdots (\boldsymbol{F}_{i-1}^B)^T & (\boldsymbol{F}_{i+1}^B)^T & \cdots (\boldsymbol{F}_N^B)^T \end{bmatrix}^T \ . \tag{2.12}$$

From Eq. 2.9, we can solve for $\boldsymbol{F}_i$ as

$$\boldsymbol{F}_i = \boldsymbol{P}^{-1}(\ddot{\boldsymbol{x}}_i - \boldsymbol{Q}) \ . \tag{2.13}$$

Finally, by substituting this expression into Eq. 2.1, we obtain an equation of the form Eq. 2.2, where

$$\boldsymbol{h} = \boldsymbol{M}_a^{-1}\boldsymbol{P}^{-1}\boldsymbol{Q} \ ,$$

$$\boldsymbol{G} = \boldsymbol{M}_a^{-1} \ , \tag{2.14}$$

in which

$$\boldsymbol{M}_a = m_i\boldsymbol{I} + \boldsymbol{P}^{-1}. \tag{2.15}$$

2.1.3 Controller Design

Our objective is for a team of robots to transport the load at a constant speed v_{des} in a target direction, defined by the angle γ in Fig. 2.1. To achieve this, we will design controllers for each robot that regulate the magnitude of its velocity to v_{des} and the direction of its velocity to γ. Since the controllers for each robot will be identical, we will drop the subscript i from the robot state variables and parameters in this section. We define the state vector for a robot as $\boldsymbol{q} = [\boldsymbol{x}^T \ \dot{\boldsymbol{x}}^T]^T$, where $\boldsymbol{x} = [x \ y]^T$ is the robot's position in inertial coordinate frame I and $\dot{\boldsymbol{x}} = [\dot{x} \ \dot{y}]^T$ is its velocity in frame I. The state vector of the robot is then $\boldsymbol{q} = [x \ y \ \dot{x} \ \dot{y}]^T$.

It will be useful to define a second global coordinate frame I_γ, with axes labeled x_I^* and y_I^*, by rotating coordinate frame I by the angle γ (see Fig. 2.1). In this coordinate frame, the axis labeled x_I^* points in the target direction of transport. Denoting the rotation matrix from frame I to frame I_γ by $\boldsymbol{R}_I^{I_\gamma}$, the state vector in the coordinates of frame I_γ is given by $\boldsymbol{q}^* = \boldsymbol{R}_I^{I_\gamma}\boldsymbol{q}$. The components of this state vector are $\boldsymbol{q}^* = [(\boldsymbol{x}^*)^T \; (\dot{\boldsymbol{x}}^*)^T]^T = [x^* \; y^* \; \dot{x}^* \; \dot{y}^*]^T$. The transformed control input is $\boldsymbol{u}^* = \boldsymbol{R}_I^{I_\gamma}\boldsymbol{u}$. We can then write the robot dynamics Eq. 2.2 in the frame I_γ as

$$\ddot{\boldsymbol{x}}^* = \boldsymbol{h}^* + \boldsymbol{G}^*\boldsymbol{u}^* \; , \tag{2.16}$$

in which $\boldsymbol{h}^* = \boldsymbol{R}_I^{I_\gamma}\boldsymbol{h}$ and $\boldsymbol{G}^* = \boldsymbol{R}_I^{I_\gamma}\boldsymbol{G}\boldsymbol{R}_{I_\gamma}^I$. We denote the components of the vector $\boldsymbol{u}^* \in \mathbb{R}^2$ by $\boldsymbol{u}^* = [u_1^* \; u_2^*]^T$, where u_1^* is the robot's actuating force in the desired direction of transport and u_2^* is its actuating force normal to this direction. We will design each of these control inputs as a sliding mode controller that drives all possible robot trajectories $\boldsymbol{x}^*(t)$ to enter a *sliding manifold* in the robot's state space in finite time and remain on the manifold thereafter (Khalil, 1996). The robot exhibits a desired dynamical behavior when its state evolves along the manifold. To regulate the robot's speed to v_{des} in the desired direction of transport (along the x_I^* axis), we define a sliding manifold s_1 as

$$s_1 = \dot{x}^* - v_{des} = 0 \; . \tag{2.17}$$

To stabilize the direction of the robot's velocity to the angle γ, we define a sliding manifold s_2 that sets the component of the robot's velocity along the y_I^* axis to zero:

$$s_2 = \dot{y}^* = 0 \; . \tag{2.18}$$

Using the approach for sliding mode control design in (Khalil, 1996), we define the control laws for u_1^* and u_2^* as

$$u_1^* \;=\; -k_1 \; sgn(s_1) \; , \tag{2.19}$$

$$u_2^* \;=\; -k_2 \; sgn(s_2) \; , \tag{2.20}$$

where k_1 and k_2 are control gains. These gains must be large enough to stabilize the system on the sliding manifolds. We derive lower bounds on the gains in Section 2.1.4. To eliminate chattering on the sliding manifolds without considerably affecting the controller performance, the signum functions in these controllers can be replaced by saturation functions, as proposed in (Khalil, 1996) and (Slotine and Li, 1991).

We note that since $\boldsymbol{G}^*$ is not a diagonal matrix, u_2^* affects the motion along x_I^* in addition to y_I^*, and u_1^* influences the motion along y_I^* in addition to x_I^*. We can describe these effects as a bounded nonlinear term that is added to the vector $\boldsymbol{h}^*$ in Eq. 2.16. Since sliding mode controllers are robust to variations in the $\boldsymbol{h}^*$ term, these effects will not deteriorate the controller performance.

2.1.4 Stability Analysis

We first derive some preliminary results that we will need to prove the stability of the system driven by the sliding mode controllers.

Proposition 2.1.1. *The matrix $\boldsymbol{P}$ in Eq. 2.10 is positive definite.*

Proof. We define a matrix $\boldsymbol{P}_1$ as:

$$\boldsymbol{P}_1 = \begin{bmatrix} \boldsymbol{I} & (\hat{\boldsymbol{r}}_i^B)^T \end{bmatrix} \boldsymbol{M}_o^{-1} \begin{bmatrix} \boldsymbol{I} \\ \hat{\boldsymbol{r}}_i^B \end{bmatrix} . \tag{2.21}$$

Then, by Eq. 2.10, $\boldsymbol{P} = \boldsymbol{R}_B^I \boldsymbol{P}_1 \boldsymbol{R}_I^B$. Since rotation matrices are invertible, the matrices $\boldsymbol{P}$ and $\boldsymbol{P}_1$ are similar, and thus they have the same eigenvalues. Using the definition $\boldsymbol{r}_i^B = \begin{bmatrix} r_{i,x} & r_{i,y} \end{bmatrix}^T$ and the definition of $\boldsymbol{M}_o$ from Eq. 2.8, $\boldsymbol{P}_1$ can be calculated from Eq. 2.21 as

$$\boldsymbol{P}_1 = \begin{bmatrix} \frac{1}{m_o} + \frac{r_{i,y}^2}{I_o} & \frac{-r_{i,x} r_{i,y}}{I_o} \\ \frac{-r_{i,x} r_{i,y}}{I_o} & \frac{1}{m_o} + \frac{r_{i,x}^2}{I_o} \end{bmatrix} . \tag{2.22}$$

The eigenvalues of $\boldsymbol{P}_1$ are

$$\lambda_1 = \frac{1}{m_o}, \quad \lambda_2 = \frac{1}{m_o} + \frac{||\boldsymbol{r}_i^B||^2}{I_o}, \tag{2.23}$$

which are both positive. Since these are also the eigenvalues of $\boldsymbol{P}$, the matrix $\boldsymbol{P}$ is positive definite. □

Proposition 2.1.2. *The matrix $\boldsymbol{G}$ in Eq. 2.14, and consequently $\boldsymbol{G}^*$ in Eq. 2.16, is positive definite and has constant eigenvalues.*

Proof. Since $\boldsymbol{G} = \boldsymbol{M}_a^{-1}$ by Eq. 2.14, we need to show that $\boldsymbol{M}_a$, and consequently $\boldsymbol{M}_a^{-1}$, is positive definite with constant eigenvalues. Let $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ denote the eigenvectors of $\boldsymbol{M}_a$, with corresponding eigenvalues μ_1 and μ_2. Using Eq. 2.15 for $\boldsymbol{M}_a$, we obtain

$$\boldsymbol{M}_a \boldsymbol{e}_j = (m_i \boldsymbol{I} + \boldsymbol{P}^{-1})\boldsymbol{e}_j = \mu_j \boldsymbol{e}_j, \quad j = 1, 2. \tag{2.24}$$

This equation can be rearranged as

$$\boldsymbol{P}^{-1}\boldsymbol{e}_j = \mu_j \boldsymbol{e}_j - m_i \boldsymbol{e}_j = (\mu_j - m_i)\boldsymbol{e}_j, \quad j = 1, 2. \tag{2.25}$$

Hence, the eigenvalues of $\boldsymbol{P}^{-1}$ are $\mu_1 - m_i$ and $\mu_2 - m_i$. Since the eigenvalues of $\boldsymbol{P}$ were found to be λ_1 and λ_2 as defined in Eq. 2.23, and the eigenvalues of $\boldsymbol{P}^{-1}$ are the inverses of the eigenvalues of $\boldsymbol{P}$, we have that $\mu_1 - m_i = \lambda_1^{-1}$ and $\mu_2 - m_i = \lambda_2^{-1}$. Therefore, the eigenvalues of $\boldsymbol{M}_a$ are $\mu_1 = m_i + \lambda_1^{-1}$ and $\mu_2 = m_i + \lambda_2^{-1}$, which are both positive and constant, making $\boldsymbol{M}_a$ a positive definite matrix with constant eigenvalues. □

Lemma 2.1.3. *If all the robots in a transport team apply control forces Eq. 2.19 and Eq. 2.20 to the load, then the angular velocity of the load will remain bounded.*

Proof. Since the robots are rigidly attached to the load, the rotational dynamics of the entire system are given by

$$I_s \ddot{\theta}_o \;=\; \sum_{i=1}^{N} \hat{\boldsymbol{r}}_i^B \boldsymbol{R}_{I_\gamma}^I \boldsymbol{u}_i^*, \quad I_s = I_o + \sum_{i=1}^{N} m_i \left\| \boldsymbol{r}_i^B \right\|^2 . \tag{2.26}$$

We define the angular difference between the load's orientation and the target direction as $\phi = \theta_o - \gamma$. Since γ is constant, $\dot{\phi} = \dot{\theta}_o$ and $\ddot{\phi} = \ddot{\theta}_o$. Writing Eq. 2.26 in terms of ϕ and substituting in the control laws Eq. 2.19 and Eq. 2.20, we obtain:

$$\begin{aligned} I_s \ddot{\phi} = & \left(k_1 \sum_{i=1}^{N} r_{i,y} sgn(s_{i,1}) - k_2 \sum_{i=1}^{N} r_{i,x} sgn(s_{i,2}) \right) \cos(\phi) \\ & + \left(k_1 \sum_{i=1}^{N} r_{i,x} sgn(s_{i,1}) + k_2 \sum_{i=1}^{N} r_{i,y} sgn(s_{i,2}) \right) \sin(\phi), \end{aligned} \tag{2.27}$$

where $s_{i,1}$ and $s_{i,2}$ are the sliding modes Eq. 2.17 and Eq. 2.18 that are defined in terms of the velocity $\dot{\boldsymbol{x}}_i^* = [\dot{x}_i^* \ \dot{y}_i^*]^T$ of robot i in coordinate frame I_γ. Since $\dot{\boldsymbol{x}}_i^*$ is a function of $\dot{\boldsymbol{x}}_o = [\dot{x}_o \ \dot{y}_o]^T$, the velocity of the load's CG in coordinate frame I, and the load's orientation $\theta_o = \phi + \gamma$ and angular velocity $\dot{\theta}_o = \dot{\phi}$, we can write Eq. 2.27 in the following form:

$$\ddot{\phi} = \eta(\dot{x}_o, \dot{y}_o, \phi, \dot{\phi}) \cos(\phi) + \zeta(\dot{x}_o, \dot{y}_o, \phi, \dot{\phi}) \sin(\phi), \tag{2.28}$$

where the coefficients η and ζ are bounded since both are finite summations of signum functions:

$$|\eta| \leq \delta_\eta \ , \qquad |\zeta| \leq \delta_\zeta \ . \tag{2.29}$$

To prove the boundedness of the load's angular velocity $\dot{\phi}$ from Eq. 2.28, we can use the comparison lemma presented in (Khalil, 1996). Here, we apply this lemma to the simpler equation $\ddot{\phi} = \eta \cos(\phi)$, since a similar approach can be used for the entire Eq. 2.28. We define a function

$$v(t) = \frac{1}{2} \dot{\phi}(t)^2 \tag{2.30}$$

whose time derivative can be calculated as

$$\dot{v} \;=\; \dot{\phi}\ddot{\phi} \;=\; \dot{\phi}\eta\cos(\phi) \;\leq\; \delta_\eta\dot{\phi}\cos(\phi), \tag{2.31}$$

with the bound on η defined in Eq. 2.29. Using Eq. 2.30, we can write the upper bound in Eq. 2.31 in terms of $v(t)$ as:

$$\dot{v} \;\leq\; \delta_\eta\sqrt{2v}\cos\left(\int_0^t \sqrt{2v}d\tau\right). \tag{2.32}$$

We now define another function, $w(t)$, that is the solution to the following equation:

$$\dot{w} \;=\; \delta_\eta\sqrt{2w}\cos\left(\int_0^t \sqrt{2w}d\tau\right), \quad w(0) = w_0. \tag{2.33}$$

By the comparison lemma from Khalil (1996), we can conclude that $v(t) \leq w(t)$ for all $t \geq 0$. From the definition of $v(t)$ in Eq. 2.30, this implies that $\dot{\phi}(t)^2/2 \leq w(t)$, and thus we obtain an upper bound on the load's angular velocity,

$$|\dot{\phi}(t)| \;\leq\; \sqrt{2|w(t)|}, \quad t \geq 0. \tag{2.34}$$

We can derive an expression for the upper bound in Eq. 2.34 by solving Eq. 2.33 for $w(t)$ and then obtaining an upper bound for $|w(t)|$. To solve Eq. 2.33, we can use the following change of variables,

$$\xi = \sqrt{2w} \;\Rightarrow\; \dot{\xi} = \frac{\dot{w}}{\sqrt{2w}},$$

and rewrite Eq. 2.33 as:

$$\dot{\xi} = \delta_\eta\cos\left(\int_0^t \xi d\tau\right), \quad \xi(0) = \sqrt{2w_0}. \tag{2.35}$$

Using another change of variables,

$$\psi = \int_0^t \xi d\tau \;\Rightarrow\; \dot{\psi} = \xi,\; \ddot{\psi} = \dot{\xi},$$

Eq. 2.35 can be written as

$$\ddot{\psi} = \delta_\eta \cos(\psi), \quad \psi(0) = \psi_0, \; \dot{\psi}(0) = \sqrt{2w_0}. \tag{2.36}$$

This is the equation of motion of a simple pendulum, which can be integrated once to obtain

$$\frac{1}{2}\dot{\psi}^2 - \delta_\eta \sin(\psi) \;=\; \frac{1}{2}\dot{\psi}^2(0) - \delta_\eta \sin(\psi_0) \;\equiv\; c. \tag{2.37}$$

Using the fact that $\dot{\psi} = \xi = \sqrt{2w}$, we have the relation $w = \dot{\psi}^2/2$, and so by Eq. 2.37,

$$w \;=\; \delta_\eta \sin(\psi) - \delta_\eta \sin(\psi_0) + w_0. \tag{2.38}$$

Then, using the triangle inequality and the fact that $w_0 > 0$,

$$|w| \;\leq\; 2\delta_\eta + w_0.$$

If we use the same procedure for the term containing ζ in Eq. 2.28, we can modify the bound as:

$$|w| \;\leq\; 2\delta_\eta + 2\delta_\zeta + w_0. \tag{2.39}$$

Substituting Eq. 2.39 into Eq. 2.34 yields the following finite upper bound on the load's angular velocity:

$$|\dot{\phi}(t)| \leq \sqrt{2|w(t)|} \leq \sqrt{4\delta_\eta + 4\delta_\zeta + 2w_0}, \quad t \geq 0. \tag{2.40}$$

Note that since $v(0) \leq w(0) = w_0$ by the comparison lemma and $v(0) = \dot{\phi}(0)^2/2$ by Eq. 2.30, setting $w_0 = 0$ implies that $\dot{\phi}(0) = 0$, meaning that the load starts with zero angular velocity at $t = 0$. □

The nonlinear term $\boldsymbol{h}$ defined in Eq. 2.14 is a function of the load's angular velocity ω_o, mass, and geometric properties, as well as the forces applied by the robots. Lemma 2.1.3 states that ω_o is bounded, and the other parameters are bounded as well due to the fact that the load has finite mass and dimensions and the robots' forces cannot exceed a saturation limit. This implies that $\boldsymbol{h}$ is bounded, a result that we will use subsequently in our stability analysis.

To analyze the stability of the system, we follow the approach in (Khalil, 1996) and define the Lyapunov functions $V_1 = \frac{1}{2}s_1^2$ and $V_2 = \frac{1}{2}s_2^2$, which measure the distance of a robot state trajectory $\boldsymbol{x}^*$ from the sliding manifolds Eq. 2.17 and Eq. 2.18, respectively. The time derivatives of these functions are:

$$\dot{V}_1 = s_1\dot{s}_1 = s_1\ddot{x}^*, \tag{2.41}$$

$$\dot{V}_2 = s_2\dot{s}_2 = s_2\ddot{y}^*. \tag{2.42}$$

In order for the system to be asymptotically stable, these functions should both be negative whenever $|s_1|, |s_2| \neq 0$.

We will conduct the analysis just for $\dot{V}_1$, since the analysis for $\dot{V}_2$ is similar. The expression for $\ddot{x}^*$ can be obtained from the first component of Eq. 2.16 and substituted into Eq. 2.41, yielding

$$\dot{V}_1 = s_1(h_1^* + g_{11}^*u_1^* + g_{12}^*u_2^*), \tag{2.43}$$

where h_1^* is the first component of $\boldsymbol{h}^*$ and g_{ij}^* is the entry of matrix $\boldsymbol{G}^*$ at row i and column j. Our finding that $\boldsymbol{h}$ is bounded implies that $h_1^* < \rho_1$ for some positive constant ρ_1. In addition, suppose that ϵ_{11}, ϵ_{12}, and g_0 are positive constants such that $g_{11}^* \geq g_0 > 0$ and the following condition is satisfied:

$$\left|\frac{h_1^* + g_{12}^*u_2^*}{g_{11}^*}\right| \leq \rho_1 + \epsilon_{11}|u_1^*| + \epsilon_{12}|u_2^*|. \tag{2.44}$$

Here, the constants ϵ_{ij} incorporate the uncertainties associated with $(g_{12}^*u_2^*)/g_{11}^*$.

By Proposition 2.1.2, the matrix $\boldsymbol{G}^*$ is positive definite, which implies that its diagonal elements g_{ii}^* are positive. Therefore, we can multiply both sides of Eq. 2.44

by $g_{11}^* \left| s_1 \right|$ to obtain:

$$\begin{aligned} |s_1|(h_1^* + g_{12}^* u_2^*) &\leq \rho_1 g_{11}^* \left| s_1 \right| + g_{11}^* \epsilon_{11} \left| u_1^* \right| \left| s_1 \right| \\ &\quad + g_{11}^* \epsilon_{12} \left| u_2^* \right| \left| s_1 \right| . \end{aligned} \tag{2.45}$$

Noting that $s_1 \leq |s_1|$ and that $|u_1^*| = k_1$, $|u_2^*| = k_2$ by Eq. 2.19 and Eq. 2.20, we have that:

$$\begin{aligned} s_1(h_1^* + g_{12}^* u_2^*) &\leq \rho_1 g_{11}^* \left| s_1 \right| + g_{11}^* \epsilon_{11} k_1 \left| s_1 \right| \\ &\quad + g_{11}^* \epsilon_{12} k_2 \left| s_1 \right| . \end{aligned} \tag{2.46}$$

If we add $s_1 g_{11}^* u_1^*$ to both sides of this inequality, the term on the left side becomes $\dot{V}_1$ by Eq. 2.43. Then, noting that $s_1 g_{11}^* u_1^* = -g_{11}^* k_1 \left| s_1 \right|$, this inequality can be written as

$$\dot{V}_1 \leq g_{11}^* \left| s_1 \right| (\rho_1 + \epsilon_{11} k_1 - k_1 + \epsilon_{12} k_2) \tag{2.47}$$

We can follow the same procedure to compute an upper bound on $\dot{V}_2$.

From Eq. 2.47, we can derive the following condition on ρ_1 to ensure that $\dot{V}_1 < 0$ whenever $\left| s_1 \right|, \left| s_2 \right| \neq 0$:

$$\rho_1 \leq (1 - \epsilon_{11}) k_1 - \epsilon_{12} k_2. \tag{2.48}$$

Similarly, the following condition on a positive constant ρ_2 can be derived to ensure that $\dot{V}_2 < 0$ whenever $\left| s_1 \right|, \left| s_2 \right| \neq 0$:

$$\rho_2 \leq (1 - \epsilon_{22}) k_2 - \epsilon_{21} k_1, \tag{2.49}$$

where ϵ_{21}, ϵ_{22} are positive constants. Now, by defining $\boldsymbol{b} = [b_1 \; b_2]^T$ in which $b_1, b_2 > 0$, the inequalities Eq. 2.48 and Eq. 2.49 can be written in the form of a matrix equation as:

$$\begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \left(\boldsymbol{I} - \begin{bmatrix} \epsilon_{11} & \epsilon_{12} \\ \epsilon_{21} & \epsilon_{22} \end{bmatrix} \right) \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} . \tag{2.50}$$

Define the matrix with entries ϵ_{ij} as $\boldsymbol{E}$. Since k_i, ρ_i, and b_i are positive, $(\boldsymbol{I}-\boldsymbol{E})$ must be nonsingular, and all elements of $(\boldsymbol{I}-\boldsymbol{E})^{-1}$ must be positive. This implies that $(\boldsymbol{I}-\boldsymbol{E})$ is an M-matrix (Khalil, 1996). Then, Eq. 2.50 can be solved for the control gains k_1 and k_2:

$$\begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = (\boldsymbol{I}-\boldsymbol{E})^{-1} \begin{bmatrix} \rho_1 + b_1 \\ \rho_2 + b_2 \end{bmatrix}. \tag{2.51}$$

By choosing these control gains and substituting them into the upper bound Eq. 2.47 on $\dot{V}_1$ and the corresponding upper bound on $\dot{V}_2$, we obtain the following inequalities:

$$\dot{V}_1 \leq -g_0 b_1 \left| s_1 \right|, \quad \dot{V}_2 \leq -g_0 b_2 \left| s_2 \right|. \tag{2.52}$$

Since $g_0, b_1, b_2 > 0$, it is evident that $\dot{V}_1 < 0$ and $\dot{V}_2 < 0$ when $\left| s_1 \right|, \left| s_2 \right| \neq 0$. Hence, the system is asymptotically stable for these gains k_1 and k_2, meaning that all state trajectories will reach the intersection of the two sliding manifolds in finite time and remain on it thereafter.

2.1.5 Simulation Results

We validated our sliding mode control strategies with simulations of point-mass robots in MATLAB and with high-fidelity 3D physics simulations in the robot simulator Webots (Michel, 2004). The robots in the Webots simulations are 3D models of the small mobile robot platform *Pheeno* that has been developed in our lab (Wilson *et al.*, 2016). To address the problem of chattering on the sliding manifolds, we used the approach mentioned in Section 2.1.3: we replaced each signum function $sgn(s_i)$ in the controllers Eq. 2.19 and Eq. 2.20 with a saturation function $sat(s_i/\epsilon_{b_i})$, where ϵ_{b_i} is a *boundary layer parameter* that gives the bounds on an envelope around $s_i = 0$ within which trajectories can evolve to avoid chattering.

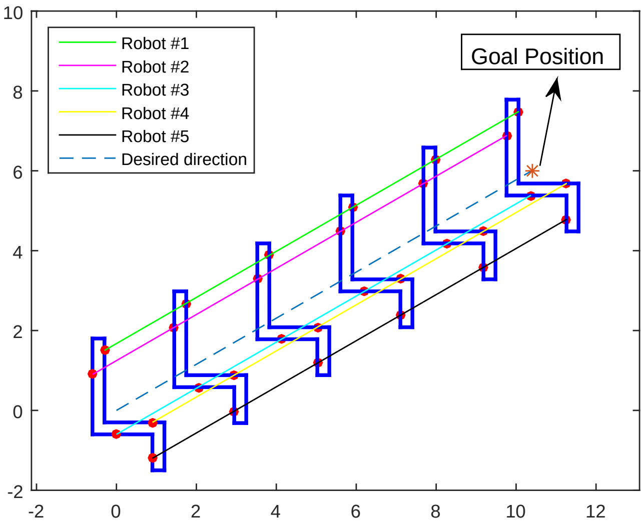


Figure 2.2: Collective Transport of a Load by Five Point-Mass Robots.

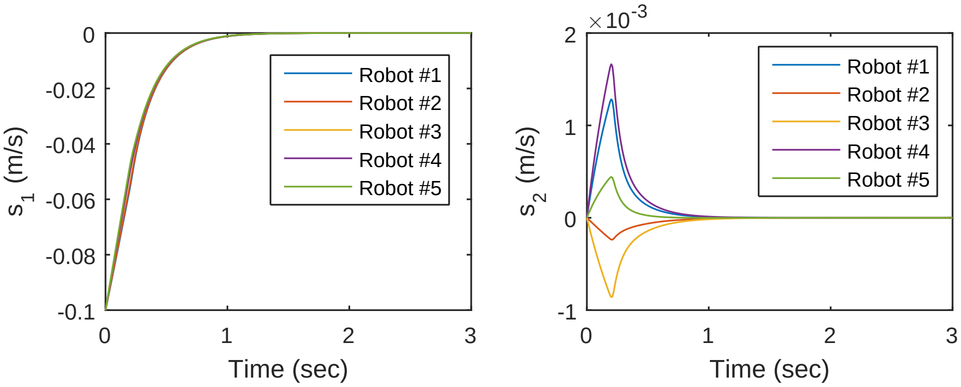


Figure 2.3: Sliding Mode Parameters (s_1 and s_2) of the Five Robots.

Simulation with point-mass robots

We simulated a scenario in which five point-mass robots, marked by the red dots in Fig. 2.2, must transport an asymmetric load to a goal, the heading to which is $\gamma = 30°$. The desired load velocity was set to $v_{des} = 0.1\, m/s$, and the controller parameters were set to $k_1 = k_2 = 0.4$, $\epsilon_{b_1} = \epsilon_{b_2} = 0.01$. The mass of the load is $1\, kg$, and its moment of inertia is $0.33\, kg/m^2$. Each robot has a mass of $0.1\, kg$ and can apply a maximum force of $0.1\, N$ on the load. The system was simulated for 120 s. As Fig. 2.2 shows, the load and the robots exhibit fairly straight trajectories that are

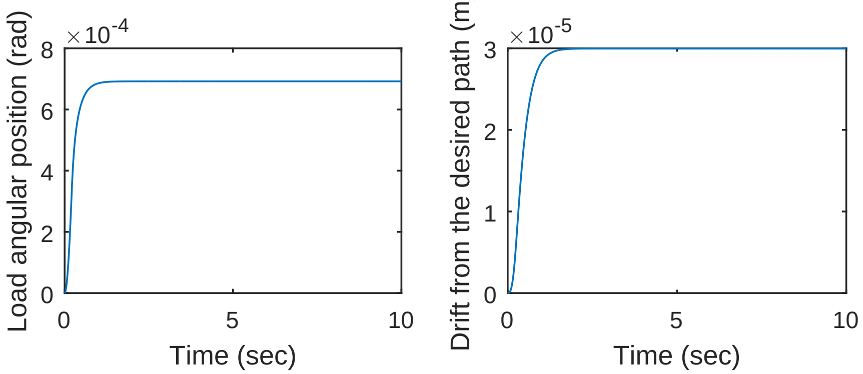


Figure 2.4: Load Rotation (Left) and its Drift from the Desired Path (Right).

parallel to the desired path to the goal, illustrated by the dashed line. Also, Fig. 2.3 plots the values of the sliding mode parameters, s_1 and s_2, for all the robots during the first 3 s of the transport. These values all quickly converge to the boundary layer ($|s_1| < 0.01$, $|s_2| < 0.01$) within $1\,s$. Fig. 2.4, which plots the load's angular position and its drift from the desired path, shows that system converges to a stable equilibrium state after a negligible initial load rotation that produces a slight initial drift of about $0.03\,mm$.

Simulations with a model of Pheeno in Webots

We also developed 3D simulations that incorporate realistic physical effects arising from the robots' wheeled actuation system and the additional degrees of freedom introduced by the manipulator arms. In addition, these simulations required modifications to the sliding mode controllers to account for the fact that Pheeno is a nonholonomic, differential-drive platform. As defined, the controllers require the velocity of the *attachment point* of a robot to the load, and so they require the velocity of Pheeno's end-effector, which necessitates computing the Jacobian matrix and consequently including the geometry of the manipulator arm in the control commands. However, there is an alternate way to control the heading and velocity of Pheeno during transport, which we pursue here. Let $\dot{\theta}_R$ and $\dot{\theta}_L$ denote the angular velocities of the right and left wheel of Pheeno, respectively, and τ_R and τ_L be the corresponding actuation torques on the wheels. These torques are the control inputs to the robot. We define:

$$\begin{aligned} \dot{\theta}_H &= \frac{1}{2}(\dot{\theta}_R - \dot{\theta}_L), \;\; \tau_H = \frac{1}{2}(\tau_R - \tau_L), \\ \dot{\theta}_V &= \frac{1}{2}(\dot{\theta}_R + \dot{\theta}_L), \;\; \tau_V = \frac{1}{2}(\tau_R + \tau_L). \end{aligned} \tag{2.53}$$

We diagonalize the linear model developed in (Ivanjko *et al.*, 2010) for a differential-drive robot and write it in the following form:

$$\begin{bmatrix} A-B & 0 \\ 0 & A+B \end{bmatrix} \begin{bmatrix} \ddot{\theta}_H \\ \ddot{\theta}_V \end{bmatrix} + \begin{bmatrix} K & 0 \\ 0 & K \end{bmatrix} \begin{bmatrix} \dot{\theta}_H \\ \dot{\theta}_V \end{bmatrix} = \begin{bmatrix} \tau_H \\ \tau_V \end{bmatrix}, \tag{2.54}$$

where the constants A and B depend on the geometry and mass properties of the robot, and the constant K is the damping in the wheels. Eq. 2.54 provides us with two decoupled equations. One equation governs the robot's heading angle φ, which is proportional to $(\theta_R - \theta_L)$, and the other governs the robot's speed v, which is proportional to $(\dot{\theta}_R + \dot{\theta}_L)$. Defining $s_H = \varphi - \gamma$ and $s_V = v - v_{des}$, we can formulate the following sliding mode controllers for the robot's heading and speed:

$$\tau_H = -k_H \; sat\left(\frac{s_H}{\epsilon_{b_H}}\right), \quad \tau_V = -k_V \; sat\left(\frac{s_V}{\epsilon_{b_V}}\right). \tag{2.55}$$

We implemented these controllers in a Webots simulation in which five Pheeno robots grasp a load, lift it simultaneously, and transport it to a goal location at a heading of $\gamma = 30°$. The desired load velocity, load mass and moment of inertia, robot mass, and robot maximum force were all set to the same values as in the point-mass simulations. The system was simulated for 120 s. The controller parameters were set to $k_H = 0.03$, $k_V = 0.09$, $\epsilon_{b_H} = 0.01$, and $\epsilon_{b_V} = 0.1$. Snapshots of the simulation are shown in Fig. 2.5. Fig. 2.6 plots the load and the robot trajectories, which are straight and parallel in the desired direction. Fig. 2.7 shows the time evolution of the sliding mode parameters, which all converge within the specified boundary layers.

2.1.6 Experimental Results

To further validate the control strategies, we conducted five experimental trials of collective transport with four Pheeno robots and a rectangular load. The robots and load were marked with 2D binary identification tags to enable real-time tracking of their positions and orientations by an overhead camera. The robots were initially placed in the configuration shown in Fig. 2.8. This configuration was chosen to minimize unwanted effects such as wheel slip and unnecessary stress on the central servo, which controls the yaw angle of the manipulator arm about the central axis of the robot. Each robot updated its state estimate using a basic complementary filter acting on its onboard encoders, compass, and accelerometer.

We implemented controllers similar to those in Eq. 2.55 on the robots. However,

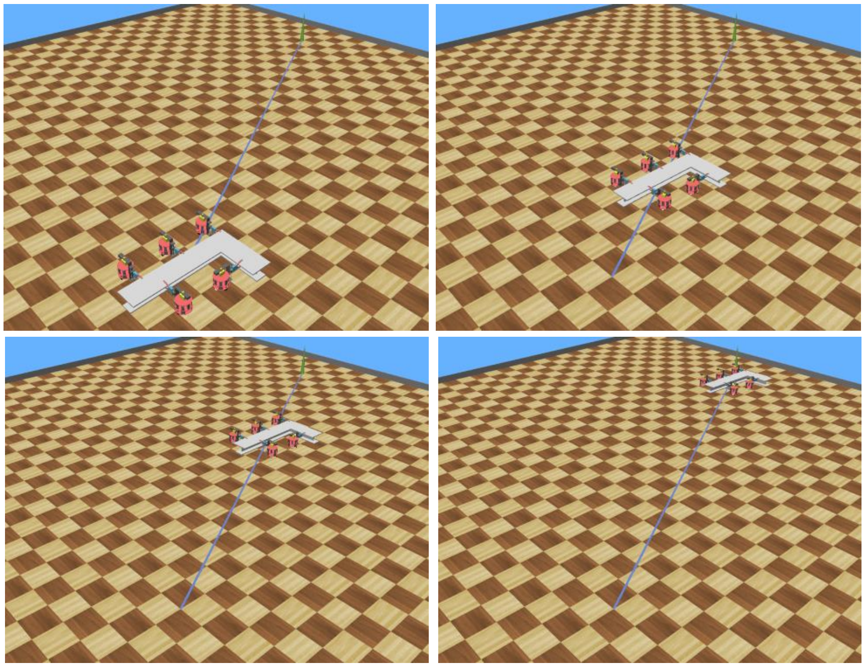


Figure 2.5: Collective Transport by Five Pheenos Simulated in Webots.

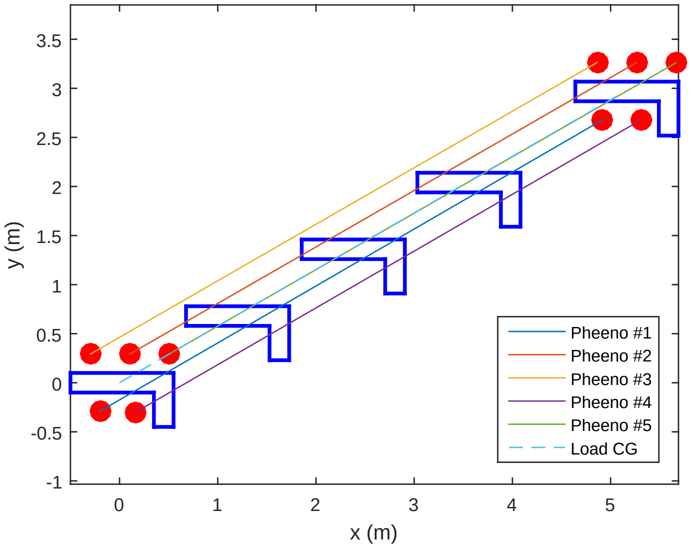


Figure 2.6: The Trajectories of the Pheenos and the Load During Transport.

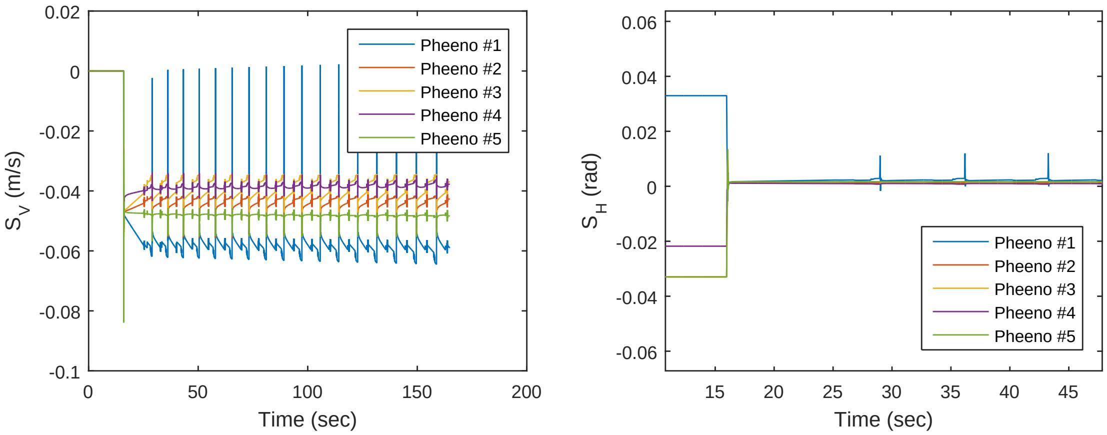


Figure 2.7: The Sliding Mode Parameters (s_V and s_H) of the Pheenos During Transport.

instead of using torque inputs, the individual motor accelerations were controlled directly, *i.e.* τ_H and τ_V in Eq. 2.55 were replaced by $\ddot{\theta}_H$ and $\ddot{\theta}_V$, respectively. Thus, the controllers required measurements of the wheel velocities and the robots' heading. The control parameters were set to $k_H = 0.01$, $k_V = 0.05$, and $\epsilon_{b_H} = \epsilon_{b_V} = 0.01$; the gains were lower than the gains in the Webots simulation to avoid causing the motors to accelerate too quickly, which results in wheel slip and odometry drift.

The robots were tasked with transporting the load at a desired velocity of $v_{des} =$

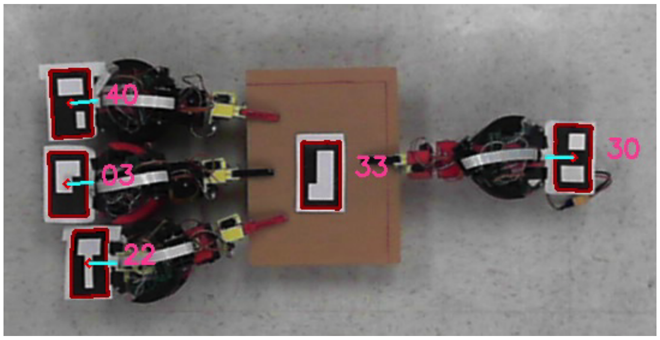


Figure 2.8: A Zoomed-in Screenshot from the Overhead Camera That Tracks the Load and Robots During the Experiment.

10 cm/s along the x-axis of the global frame defined by the overhead camera. Each trial was run for 30 s. Fig. 2.9 shows the paths of the load and transporting robots during a single experiment, and Fig. 2.10 plots the average and standard deviation of the load's velocity, heading, and trajectory over the five experiments. These plots show that the sliding mode controllers are fairly successful at achieving the control objectives. The slight rotation of the load and its deviation from the desired path in Fig. 2.9, as well as the increasing standard deviations in the plots in Fig. 2.10, are due to unavoidable drift in the onboard odometry caused by wheel slip, sensor noise, and model error, among other factors. Sensor noise can result in discrepancies in the robots' velocities, causing the robots to exert torques on each other through the load, which produces wheel slip and error in the odometry.

2.2 Decentralized Proportional-Integral Control

In this section, we investigate the stability properties of a proportional control scheme that is a modification of the one presented in (Rubenstein *et al.*, 2013). The paper (Rubenstein *et al.*, 2013) proposes a decentralized approach to the problem that we address, in which each robot applies a force to the payload that is defined as a proportional velocity controller. It is demonstrated that the payload moves in a straight line toward the goal with no more than 180° of rotation. However, there is

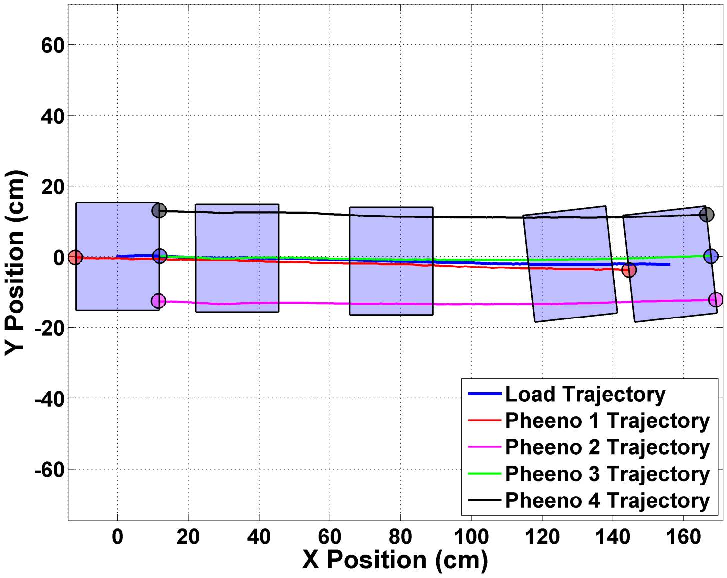


Figure 2.9: The Trajectories of the Pheenos and the Transported Load During One Experiment. The Rectangle Shows the Orientation of the Load at Several Time Points. The Colored Circles Mark the Robot Attachment Points at the Beginning and End of the Experiment.

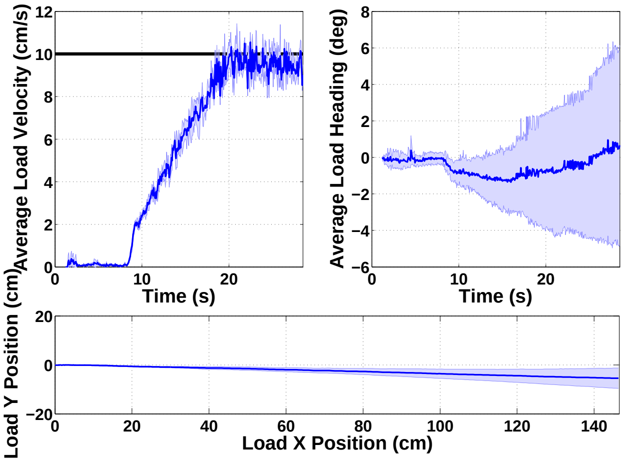


Figure 2.10: The States of the Load During the Transport Experiments. The Dark Blue Lines Are the Mean States Averaged over Five Trials, and the Light Blue Area Shows the Standard Deviation. *Top Left:* The Average Velocity of the Load During Transport. *Top Right:* The Average Heading of the Load During Transport. *Bottom:* The Average Trajectory of the Load During Transport.

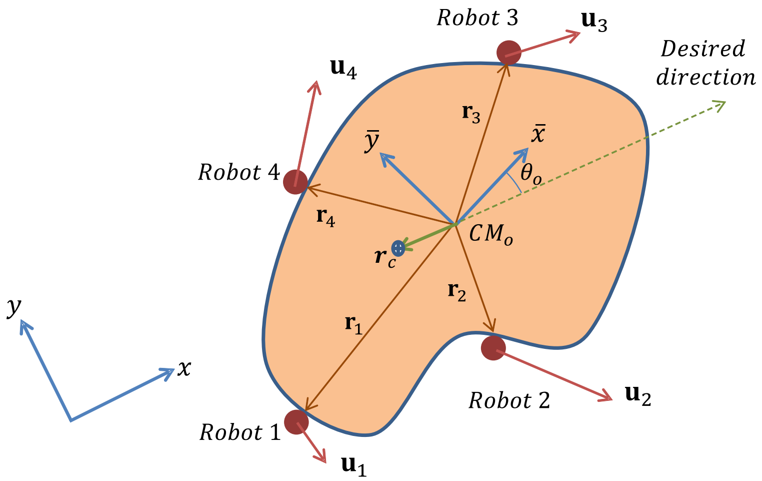


Figure 2.11: Illustration of a Collective Transport Team with Four Point-Mass Robots and the Associated Coordinate Systems.

no stability analysis that guarantees the convergence of the system's dynamics to the desired motion. In our control scheme, the robots also exert a force component perpendicular to the target direction, in order to maintain a straight transport trajectory. We prove that the closed-loop system is exponentially stable with this controller, and we characterize the rate of the system's convergence to the target transport velocity in terms of the robots' distribution around the payload. In addition, we introduce an integral controller to drive any drift of the payload from the desired path to zero. We note that our analysis on pure proportional control also applies to our previous decentralized strategy for collective transport (**Farivarnejad, H.** *et al.*, 2016), during the phase when the system trajectories enter the boundary layer defined in the sliding mode controllers.

2.2.1 Dynamical Model

We model the dynamics of the system in Fig. 2.11, a load that is transported by N point-mass robots, which we studied in our previous work (**Farivarnejad, H.** *et al.*, 2016). Here, we derive the equations of motion for the entire system, comprised of both the load and the robots, whereas in (**Farivarnejad, H.** *et al.*, 2016) we derived the dynamics of each robot. We define m_r as the mass of each robot, m_o as the mass of the payload, and I_o as the load's moment of inertia about the axis normal to the plane of the motion and passing through its center of gravity. We also define $\boldsymbol{r}_c$ as the vector from the center of mass of the entire system (CM_o) to the load's center of mass, and $\boldsymbol{r}_i$ as the vector from CM_o to the attachment point of robot i. Both $\boldsymbol{r}_c$ and $\boldsymbol{r}_i$ are expressed in the inertial reference frame shown in Fig. 2.11, defined such that the x-axis points in the target direction of transport. Then, the mass m and moment of inertia I of the entire system are given by:

$$\begin{aligned} m &= m_o + N m_r, \\ I &= I_o + m_o \left\| \boldsymbol{r}_c \right\|^2 + m_r \sum_{i=1}^{N} \left\| \boldsymbol{r}_i \right\|^2 . \end{aligned} \tag{2.56}$$

Each robot i applies an actuating force $\boldsymbol{u}_i = [u_{i,x} \; u_{i,y}]^T$ to the payload. We denote the vector of all applied forces by $\boldsymbol{u} = \begin{bmatrix} (\boldsymbol{u}_1)^T \cdots (\boldsymbol{u}_N)^T \end{bmatrix}^T$. We define the position of CM_o in the inertial reference frame as $\boldsymbol{x}_o = [x_o \; y_o]^T$ and the load's orientation in this frame as θ_o. We will use $\boldsymbol{q}_o = [x_o \; y_o \; \theta_o]^T$ as the generalized coordinates that describe the motion of the entire system. Then we can write the equation of motion of the system as:

$$\begin{bmatrix} m\boldsymbol{I} & 0 \\ 0 & I \end{bmatrix} \ddot{\boldsymbol{q}}_o = \begin{bmatrix} \boldsymbol{I} & \cdots & \boldsymbol{I} \\ \hat{\boldsymbol{r}}_1 & \cdots & \hat{\boldsymbol{r}}_N \end{bmatrix} \boldsymbol{u}, \tag{2.57}$$

where $\boldsymbol{I}$ is the identity matrix and $\hat{\boldsymbol{r}}$ is a skew-symmetric matrix defined by $\boldsymbol{r}_i \times \boldsymbol{u} = \hat{\boldsymbol{r}}_i \boldsymbol{u}$. Let $\dot{x}_i$ and $\dot{y}_i$ be the speed of robot i along the x and y axes of the inertial frame. We define the components of $\boldsymbol{u}_i$ for each robot i as proportional velocity controllers:

$$u_{i,x} = k(v_{des} - \dot{x}_i), \quad u_{i,y} = k(-\dot{y}_i), \tag{2.58}$$

where k is the controller gain and v_{des} is the desired transport speed. This controller drives each robot's velocity to v_{des} along the desired direction of transport, with no velocity component perpendicular to this direction. When all robots attain this velocity, the load moves in the target direction at speed v_{des} with zero angular velocity. Using the kinematic equations of the payload, we can obtain expressions for $\dot{x}_i$ and $\dot{y}_i$ in terms of $\dot{\boldsymbol{q}}_o$ and then rewrite Eq. 2.58 as:

$$\begin{aligned} u_{i,x} &= k\left(v_{des} - \dot{x}_o + \dot{\theta}_o||\boldsymbol{r}_i||\sin(\theta_o + \theta_i)\right), \\ u_{i,y} &= k\left(-\dot{y}_o - \dot{\theta}_o||\boldsymbol{r}_i||\cos(\theta_o + \theta_i)\right), \end{aligned} \tag{2.59}$$

where θ_i is the angle of vector $\boldsymbol{r}_i$ with respect to a local coordinate frame fixed to the load, as shown in Eq. 2.12. Substituting the controller in Eq. 2.59 for each robot into Eq. 2.57, we obtain the following equations for the closed-loop system:

$$\begin{aligned} m\ddot{x}_o &= k\left(N(v_{des} - \dot{x}_o) + \dot{\theta}_o\sum_{i=1}^{N}||\boldsymbol{r}_i||\sin(\theta_o + \theta_i)\right), \\ m\ddot{y}_o &= k\left(-N\dot{y}_o - \dot{\theta}_o\sum_{i=1}^{N}||\boldsymbol{r}_i||\cos(\theta_o + \theta_i)\right), \\ I\ddot{\theta}_o &= -k\sum_{i=1}^{N}\left(\dot{y}_o||\boldsymbol{r}_i||\cos(\theta_o + \theta_i) + \dot{\theta}_o||\boldsymbol{r}_i||^2\cos^2(\theta_o + \theta_i)\right) \\ &+ k\sum_{i=1}^{N}(\dot{x}_o - v_{des})||\boldsymbol{r}_i||\sin(\theta_o + \theta_i) - k\sum_{i=1}^{N}\dot{\theta}_o||\boldsymbol{r}_i||^2\sin^2(\theta_o + \theta_i). \end{aligned} \tag{2.60}$$

Defining $s_x \equiv \dot{x}_o - v_{des}$ and $s_y \equiv \dot{y}_o$, the closed-loop dynamics in Eq. 2.60 can be rewritten in the following compact form:

$$\begin{aligned} &m\dot{s}_x + c_t s_x - k f_s(\theta_o)\dot{\theta}_o = 0, \\ &m\dot{s}_y + c_t s_y + k f_c(\theta_o)\dot{\theta}_o = 0, \\ &I\ddot{\theta}_o + c_r\dot{\theta}_o - k f_s(\theta_o)s_x + k f_c(\theta_o)s_y = 0, \end{aligned} \tag{2.61}$$

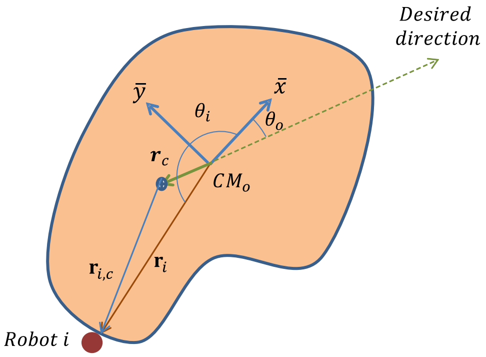


Figure 2.12: Illustration of the Geometric Parameters that Express the Position of a Robot in the Local Coordinate Frame of the Load.

where $c_t = kN$, $c_r = k\sum_{i=1}^{N} ||\boldsymbol{r}_i||^2$, and:

$$f_s(\theta_o) = \sum_{i=1}^{N} ||\boldsymbol{r}_i|| \sin(\theta_o + \theta_i),$$
$$f_c(\theta_o) = \sum_{i=1}^{N} ||\boldsymbol{r}_i|| \cos(\theta_o + \theta_i). \tag{2.62}$$

2.2.2 Stability Analysis

In this section, we characterize the stability of the equilibria of the closed-loop system in Eq. 2.61. Defining $\boldsymbol{z} = [s_x \ s_y \ \theta_o \ \dot{\theta}_o]^T$ as the state vector, we find that the system has no isolated equilibrium point and that the set $\mathcal{M}$, defined as:

$$\mathcal{M} = \left\{ \boldsymbol{z} \in \mathbb{R}^4 | s_x, s_y, \dot{\theta}_o = 0 \right\}, \tag{2.63}$$

is a continuum of equilibrium points, *i.e.* an invariant set. When the system state is in this set, the payload moves directly to the goal along a straight path ($s_y = 0$) at a regulated velocity ($s_x = 0$) with no rotational motion ($\dot{\theta}_o = 0$). The following theorem characterizes the convergence of the system trajectories to $\mathcal{M}$.

Theorem 2.2.1. *The trajectories of the system in Eq. 2.57 with the decentralized controllers in Eq. 2.58 exponentially converge to the set $\mathcal{M}$.*

Proof. We consider the following Lyapunov function:

$$V = \frac{1}{2}m(s_x^2 + s_y^2) + \frac{1}{2}I\dot{\theta}_o^2. \tag{2.64}$$

The time derivative of this function is:

$$\dot{V} = -c_t s_x^2 - c_t s_y^2 - c_r\dot{\theta}_o^2 + 2k(f_s(\theta_o)s_x - f_c(\theta_o)s_y)\dot{\theta}_o. \tag{2.65}$$

Defining $\boldsymbol{z}_1 = [s_x \ s_y \ \dot{\theta}_o]^T$, Eq. 2.65 can be written as

$$\dot{V} = -\boldsymbol{z}_1^T\boldsymbol{\Lambda}\boldsymbol{z}_1, \tag{2.66}$$

in which

$$\boldsymbol{\Lambda} = \begin{bmatrix} c_t & 0 & kf_s \\ 0 & c_t & -kf_c \\ kf_s & -kf_c & c_r \end{bmatrix}. \tag{2.67}$$

To prove the convergence of the system, we need to show that $\boldsymbol{\Lambda}$ is positive definite, or equivalently, that all its eigenvalues are positive. These eigenvalues are given by:

$$\begin{aligned} \lambda_1 &= c_t, \\ \lambda_2 &= \frac{1}{2}\left((c_t + c_r) + \sqrt{4k^2(f_s^2 + f_c^2) + (c_t - c_r)^2}\right), \\ \lambda_3 &= \frac{1}{2}\left((c_t + c_r) - \sqrt{4k^2(f_s^2 + f_c^2) + (c_t - c_r)^2}\right). \end{aligned} \tag{2.68}$$

Since c_t and c_r are strictly positive numbers, we can conclude that λ_1 and λ_2 are strictly positive as well. Therefore, we only have to determine the sign of λ_3. We first investigate the term $\xi \equiv \sqrt{f_s^2 + f_c^2}$.

Proposition 2.2.2. *For the system described by Eq. 2.57 with the robot controllers 2.58, ξ is a constant that is equal to $||\sum_{i=1}^{N}\boldsymbol{r}_i||$.*

Proof. Using Eq. 2.62, we can write:

$$\xi^2 = \sum_{i=1}^{N} ||\boldsymbol{r}_i||^2 + \sum_{i=1}^{N}\sum_{j\neq i}^{N} ||\boldsymbol{r}_i|| ||\boldsymbol{r}_j|| \cos(\theta_o + \theta_i)\cos(\theta_o + \theta_j)$$
$$+ \sum_{i=1}^{N}\sum_{j\neq i}^{N} ||\boldsymbol{r}_i|| ||\boldsymbol{r}_j|| \sin(\theta_o + \theta_i)\sin(\theta_o + \theta_j). \qquad (2.69)$$

Combining the second two terms on the right-hand side, we obtain:

$$\xi^2 = \sum_{i=1}^{N} ||\boldsymbol{r}_i||^2 + \sum_{i=1}^{N}\sum_{j\neq i}^{N} ||\boldsymbol{r}_i|| ||\boldsymbol{r}_j|| \cos(\theta_i - \theta_j). \qquad (2.70)$$

Denoting the components of $\boldsymbol{r}_i$ in the local coordinate frame as $\bar{x}_i$ and $\bar{y}_i$, we have that $\bar{x}_i = ||\boldsymbol{r}_i|| \cos(\theta_i)$ and $\bar{y}_i = ||\boldsymbol{r}_i|| \sin(\theta_i)$. We can then rewrite Eq. 2.70 as:

$$\xi^2 = \left(\sum_{i=1}^{N} (\bar{x}_i^2 + \bar{y}_i^2) + \sum_{i=1}^{N}\sum_{j\neq i}^{N} (\bar{x}_i\bar{x}_j + \bar{y}_i\bar{y}_j) \right). \qquad (2.71)$$

Finally, by separating the x and y components, we can write:

$$\xi^2 = \left((\bar{x}_1 + \cdots + \bar{x}_N)^2 + (\bar{y}_1 + \cdots + \bar{y}_N)^2 \right), \qquad (2.72)$$

which implies that $\xi = || \sum_{i=1}^{N} \boldsymbol{r}_i ||$. □

Now, we can analyze the sign of λ_3.

Proposition 2.2.3. *For a transport team with a fixed number of robots and a fixed configuration on the load, λ_3 in Eq. 2.68 is strictly positive.*

Proof. First, we calculate the critical value of ξ, defined as ξ_{cr}, at which λ_3 becomes zero. If we can show that ξ is always less than this value, then we can conclude that λ_3 is always positive. From Eq. 2.68, we calculate ξ_{cr} as:

$$\xi_{cr}^2 = N \sum_{i=1}^{N} ||\boldsymbol{r}_i||^2. \qquad (2.73)$$

From the triangle inequality, we know that $||\sum_{i=1}^{N}\boldsymbol{r}_i|| \leq \sum_{i=1}^{N}||\boldsymbol{r}_i||$, and by squaring both sides of this inequality, we have:

$$||\sum_{i=1}^{N}\boldsymbol{r}_i||^2 \ \leq \ \left(\sum_{i=1}^{N}||\boldsymbol{r}_i||\right)^2. \tag{2.74}$$

From the Cauchy-Schwarz inequality Horn and Johnson (2012), we know that

$$\left(\sum_{i=1}^{N}||\boldsymbol{r}_i||\right)^2 \ \leq \ N\sum_{i=1}^{N}||\boldsymbol{r}_i||^2. \tag{2.75}$$

which means that $\xi^2 \leq \xi_{cr}^2$ and consequently, $\xi \leq \xi_{cr}$. Excluding physically impossible configurations in which all robots occupy a single point on the perimeter of the load, which results in $\xi = \xi_{cr}$, λ_3 is strictly positive. □

Since all the eigenvalues of $\boldsymbol{\Lambda}$ are positive, it is positive definite. Furthermore, the Lyapunov function in Eq. 2.64 can be written in the quadratic form $V = \boldsymbol{z}_1^T\boldsymbol{\Delta}\boldsymbol{z}_1$, where:

$$\boldsymbol{\Delta} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix}. \tag{2.76}$$

Then, we have the following inequalities (Khalil, 1996):

$$\lambda_{min}(\boldsymbol{\Delta})||\boldsymbol{z}_1||^2 \ \leq \ V(\boldsymbol{z}_1) \ \leq \ \lambda_{max}(\boldsymbol{\Delta})||\boldsymbol{z}_1||^2 \tag{2.77}$$

In addition, using Eq. 2.66, the following upper bound can be established for $\dot{V}$:

$$\dot{V}(\boldsymbol{z}_1) \ \leq \ -\lambda_{min}(\boldsymbol{\Lambda})||\boldsymbol{z}_1||^2 \tag{2.78}$$

Therefore, we can write:

$$\dot{V} \ \leq \ -\frac{\lambda_{min}(\boldsymbol{\Lambda})}{\lambda_{max}(\boldsymbol{\Delta})}V, \tag{2.79}$$

and by *Theorem 4.10* in (Khalil, 1996), we can conclude that trajectories of the system in Eq. 2.61 exponentially converge to the invariant set $\mathcal{M}$. This completes the proof of Theorem 2.2.1. □

2.2.3 Convergence Analysis

Given the exponential stability of the closed-loop system, we can describe the convergence of its trajectories in a qualitative fashion using an exponential function that gives the lowest possible rate of convergence to $\mathcal{M}$. According to *Theorem 4.10* in Khalil (1996), the following inequality holds:

$$||\boldsymbol{z}_1(t)|| \ \leq \ b||\boldsymbol{z}_1(t_0)||\, e^{-\epsilon t}, \quad t \geq t_0, \tag{2.80}$$

where $b = \sqrt{\frac{\lambda_{max}(|\boldsymbol{Delta})}{\lambda_{min}(\boldsymbol{\Delta})}}$ and

$$\epsilon = \frac{\lambda_{min}(\boldsymbol{\Lambda})}{\lambda_{max}(\boldsymbol{\Delta})}. \tag{2.81}$$

Thus, ϵ bounds the convergence rate of the system trajectories. We now show how ϵ can be characterized in terms of the distribution of the robots around the load. Toward this end, we first determine $\lambda_{min}(\boldsymbol{\Lambda})$.

Proposition 2.2.4. *For a transport team with a fixed number of robots and a fixed configuration on the load, $\lambda_{min}(\boldsymbol{\Lambda}) = \lambda_3$, defined in Eq. 2.68.*

Proof. From Eq. 2.68, we see that $\lambda_3 < \lambda_2$. Therefore, we only need to compare λ_1 and λ_3. We rewrite λ_3 as:

$$\lambda_3 = \frac{1}{2}(c_t + c_r) - \frac{1}{2}(|c_t - c_r| + 2\varrho), \tag{2.82}$$

where ϱ is a positive number that monotonically increases with ξ. In this expression, if $c_r \leq c_t$, then $\lambda_3 = c_r - \varrho$, which is clearly smaller than c_r. Hence, in this case, $\lambda_3 \leq c_r \leq c_t = \lambda_1$. If $c_t \leq c_r$, then $\lambda_3 = c_t - \varrho$, which is clearly smaller than c_t. Thus, in this case, $\lambda_3 \leq c_t = \lambda_1$ as well. Therefore, we can conclude that $\lambda_{min}(\boldsymbol{\Lambda}) = \lambda_3$. □

To simplify our subsequent calculations, we replace the vector $\boldsymbol{z}_1$ in our stability analysis with $\boldsymbol{z}_2 = [s_x \ s_y \ r_g\dot{\theta}_o]^T$, where r_g is the radius of gyration of the system,

$$r_g = \left(\frac{I_o + m_o \left\|\boldsymbol{r}_c\right\|^2 + m_r \sum_{i=1}^{N} \left\|\boldsymbol{r}_i\right\|^2}{m_o + N m_r} \right)^{\frac{1}{2}}. \tag{2.83}$$

With this replacement, we can write the Lyapunov function in Eq. 2.64 as $V = \boldsymbol{z}_2^T \boldsymbol{\Delta}_2 \boldsymbol{z}_2$, where $\boldsymbol{\Delta}_2 = m\boldsymbol{I} \in \mathbb{R}^{3\times3}$, and $\lambda_{max}(\boldsymbol{\Delta}_2) = m$. Note that m is independent of the robot configuration on the load. We can also write Eq. 2.66 as $\dot{V} = \boldsymbol{z}_2^T \boldsymbol{\Lambda}_2 \boldsymbol{z}_2$. Setting $\lambda_{max}(\boldsymbol{\Delta}) = m$, we can calculate ϵ as:

$$\epsilon = \frac{1}{2mr_g^2}\left((r_g^2 c_t + c_r) - \sqrt{4r_g^2\xi^2 + (r_g^2 c_t - c_r)^2}\right). \tag{2.84}$$

We see that for a fixed number of robots, the distribution of the robots around the load (*i.e.*, the set of vectors $\boldsymbol{r}_i$) affects σ through the parameters r_g, c_r, and ξ. It is difficult to analyze the effect on all three parameters simultaneously. However, by following the procedure in the proof of *Proposition* 2.2.4, we can write ϵ in the following form:

$$\epsilon = \frac{k}{2mr_g^2}\left(N(r_g^2 + \rho^2 - |r_g^2 - \rho^2|) - 2\delta\right), \tag{2.85}$$

where:

$$\rho^2 = \frac{1}{N}\sum_{i=1}^{N} ||\boldsymbol{r}_i||^2, \tag{2.86}$$

and δ is a positive number that monotonically increases with $r_g\xi$.

We now show that we can derive simplified expressions for ϵ based on the relative magnitudes of r_g and ρ. From Eq. 2.83, the term $(r_g^2 - \rho^2)$ in Eq. 2.85 can be written as:

$$(r_g^2 - \rho^2) = \frac{a}{a+N}(r_{g,o}^2 + ||\boldsymbol{r}_c||^2 - \rho^2), \tag{2.87}$$

where $a = m_o/m_r$ and $r_{g,o}$ is the load's radius of gyration. Furthermore, we can write the vector $\boldsymbol{r}_c$ as:

$$\boldsymbol{r}_c = -\frac{m_r}{m_o + N m_r}\sum_{i=1}^{N} \boldsymbol{r}_{i,c} \tag{2.88}$$

where $\boldsymbol{r}_{i,c}$ is the vector from the load's center of mass to robot i (see Eq. 2.12). For each robot i, we have the relation $\boldsymbol{r}_i = \boldsymbol{r}_c + \boldsymbol{r}_{i,c}$, and by applying the triangle inequality, we have that $||\boldsymbol{r}_i|| \leq ||\boldsymbol{r}_c|| + ||\boldsymbol{r}_{i,c}||$. Squaring both sides of this inequality and summing the resulting terms over $i = 1, ..., N$, we obtain:

$$N\rho^2 \ \leq \ N||\boldsymbol{r}_c||^2 + N\rho_c^2 + 2||\boldsymbol{r}_c|| \sum_{i=1}^{N} ||\boldsymbol{r}_{i,c}||, \tag{2.89}$$

where $\rho_c^2 = \frac{1}{N}\sum_{i=1}^{N} ||\boldsymbol{r}_{i,c}||^2$. Hence, we can write:

$$-\rho_c^2 - \frac{2}{N}||\boldsymbol{r}_c|| \sum_{i=1}^{N} ||\boldsymbol{r}_{i,c}|| \ \leq \ ||\boldsymbol{r}_c||^2 - \rho^2. \tag{2.90}$$

Moreover, using the expression for $\boldsymbol{r}_c$ in Eq. 2.88 and applying Eq. 2.74 and Eq. 2.75 to the vectors $\boldsymbol{r}_{i,c}$ we can write:

$$-\rho_c^2 - \frac{2N}{a+N}\rho_c^2 \ \leq \ ||\boldsymbol{r}_c||^2 - \rho^2, \tag{2.91}$$

which yields the following lower bound for $(r_g^2 - \rho^2)$:

$$\frac{a}{a+N}\left(r_{g,o}^2 - (1 + \frac{2N}{a+N})\rho_c^2\right) \ \leq \ (r_g^2 - \rho^2). \tag{2.92}$$

Note that this lower bound is a function of a single parameter, ρ_c, that depends on the distribution of robots around the load. We can now determine the sign of $(r_g^2 - \rho^2)$ for the following two cases:

(a) $r_g < \rho$. This case happens when ρ_c is sufficiently large to make the lower bound in Eq. 2.92 a negative large number. This occurs when the robots are mostly located at positions that are far from the load's center of mass. In this scenario,

$$\epsilon = \frac{k}{m}\left(N - \frac{\delta}{r_g^2}\right). \tag{2.93}$$

This means that for a fixed number of robots, ϵ mainly depends on the value of ξ through δ.

(b) $r_g > \rho$. This case happens when ρ_c is a small number that makes the lower bound in Eq. 2.92 positive. This occurs when the robots have a uniform and close-to-symmetric distribution around the load. Under this condition,

$$\epsilon = \frac{k}{mr_g^2}(N\rho^2 - \delta). \tag{2.94}$$

Here, ϵ is not as sensitive to changes in δ (and hence ξ) as it is in the first case, since such changes could be compensated by the value of ρ^2.

2.2.4 Drift Compensation by Integral Control

When the proposed proportional controller is used, the load will inevitably drift away from the line between its initial position and its target position, which we will refer to as the *desired path*. To eliminate this drift, we add an integral term to Eq. 2.58 and modify the control law as follows:

$$\begin{aligned} u_{i,x} &= k(v_{des} - \dot{x}_i) + k_I \int_o^t (v_{des} - \dot{x}_i) d\tau, \\ u_{i,y} &= k(-\dot{y}_i) + k_I \int_o^t (-\dot{y}_i) d\tau. \end{aligned} \tag{2.95}$$

With this new controller, the closed-loop dynamics in Eq. 2.61 can be rewritten as:

$$\begin{aligned} & m\ddot{\sigma}_x + c_t\dot{\sigma}_x + k_I N \sigma_x - k f_s(\theta_o)\dot{\theta}_o - k_I \eta_s = 0, \\ & m\ddot{\sigma}_y + c_t\dot{\sigma}_y + k_I N \sigma_y + k f_c(\theta_o)\dot{\theta}_o + k_I \eta_c = 0, \\ & I\ddot{\theta}_o + c_r\dot{\theta}_o - k f_s(\theta_o)\dot{\sigma}_x + k f_c(\theta_o)\dot{\sigma}_y \\ & - k_I \left(f_s(\theta_o)\sigma_x - f_c(\theta_o)\sigma_y - \sum_{i=1}^{N} (\eta_{i,s}\dot{\eta}_{i,s} + \eta_{i,c}\dot{\eta}_{i,c}) \right) = 0, \end{aligned} \tag{2.96}$$

in which

$$\sigma_x = \int_o^t s_x d\tau, \quad \sigma_y = \int_o^t s_y d\tau, \tag{2.97}$$

and

$$\eta_s = \sum_{i=1}^{N} \eta_{i,s}, \quad \eta_c = \sum_{i=1}^{N} \eta_{i,c}, \tag{2.98}$$

where

$$\begin{aligned}\eta_{is} &= ||\boldsymbol{r}_i|| \int_o^t \sin(\theta_o + \theta_i)\dot{\theta}_o d\tau, \\ \eta_{ic} &= ||\boldsymbol{r}_i|| \int_o^t \cos(\theta_o + \theta_i)\dot{\theta}_o d\tau. \end{aligned} \tag{2.99}$$

Furthermore, noting that $\dot{\theta}_o d\tau = d\theta_o$, the two integrals in Eq. 2.99 can be calculated as:

$$\begin{aligned}\eta_{is} &= ||\boldsymbol{r}_i|| \left(\cos(\theta_o(0) + \theta_i) - \cos(\theta_o + \theta_i)\right), \\ \eta_{ic} &= ||\boldsymbol{r}_i|| \left(\sin(\theta_o + \theta_i) - \sin(\theta_o(0) + \theta_i)\right), \end{aligned} \tag{2.100}$$

where $\theta_o(0)$ is the initial orientation of the load. By substituting Eq. 2.100 into Eq. 2.96, we can rewrite the closed dynamics as:

$$\begin{aligned}& m\ddot{\sigma}_x + c_t\dot{\sigma}_x + c_I\sigma_x - \kappa\dot{\eta}_s - \kappa_I\eta_s = 0, \\ & m\ddot{\sigma}_y + c_t\dot{\sigma}_y + c_I\sigma_y + \kappa\dot{\eta}_c + \kappa_I\eta_c = 0, \\ & I\ddot{\theta}_o + c_r\dot{\theta}_o - \kappa f_s(\theta_o)\dot{\sigma}_x + \kappa f_c(\theta_o)\dot{\sigma}_y - k_I(f_s(\theta_o)\sigma_x - f_c(\theta_o)\sigma_y) + c_I\rho^2\sin(\tilde{\theta}_o) = 0, \end{aligned} \tag{2.101}$$

in which $c_I = k_I N$, and $\tilde{\theta}_o = \theta_o - \theta_o(0)$ is the difference between the current value of θ_o and its initial value, which implies $\dot{\tilde{\theta}}_o = \dot{\theta}_o$ and $\ddot{\tilde{\theta}}_o = \ddot{\theta}_o$.

This model has two more state variables than the system in Eq. 2.61 does: σ_x, which represents the accumulation of error from the desired velocity, and σ_y, which represents the drift from the desired path. Defining the new state vector as $\boldsymbol{\zeta} = [\sigma_x\ \sigma_y\ \dot{\sigma}_x\ \dot{\sigma}_y\ \theta_o\ \dot{\theta}_o]^T$, and setting $\dot{\boldsymbol{\zeta}} = \mathbf{0}$, we see that the equilibrium point of the closed-loop system is where

$$\begin{aligned}& N\sigma_x = \eta_s \\ & N\sigma_y = -\eta_c \\ & f_s(\theta_o)\sigma_x - f_c(\theta_o)\sigma_y = N\rho^2\sin(\tilde{\theta}_o). \end{aligned} \tag{2.102}$$

If we substitute the first two equations in Eq. 2.102 into the third equation and expand it, we can rewrite it as:

$$\left(\sum_{i=1}^{N}\sum_{j=1}^{N}||\boldsymbol{r}_i||||\boldsymbol{r}_j||\right)\sin(\tilde{\theta}_o) = N\rho^2\sin(\tilde{\theta}_o). \tag{2.103}$$

Separating the terms with the same indexes and the ones with different indexes in the left hand side of Eq. 2.103, and with regards to Eq. 2.86, we see that in the equilibrium point we have:

$$\left(\sum_{i=1}^{N}\sum_{j\neq i}^{N}||\boldsymbol{r}_i||||\boldsymbol{r}_j||\right)\sin(\tilde{\theta}_o) = 0. \tag{2.104}$$

Therefore, the equilibrium point is where $\tilde{\theta}_o = k\pi$ for $k \in \mathbb{Z}$, and the corresponding σ_x and σ_y are obtained from Eq. 2.102. By linearization, we can easily see that for odd values of k, the equilibrium point is unstable. However, the following theorem expresses the asymptotic stability of the trajectories of the closed-loop system in Eq. 2.96 to the equilibrium points with even values of k, where θ_o is equal to its initial value, and σ_x and σ_y are equal to zero with regards to Eq. 2.100 and Eq. 2.102. First, we state the following proposition, which will be used in the proof of the theorem.

Proposition 2.2.5. *The following function is a candidate Lyapunov function only for the equilibrium points with even values of k:*

$$W \;=\; V + \frac{1}{2}k_I\sum_{i=1}^{N}\left((\sigma_x - \eta_{i,s})^2 + (\sigma_y + \eta_{i,c})^2\right). \tag{2.105}$$

Proof. First, we can easily validate that this function is nonnegative since it is a sum of squares. Moreover, $\dot{\sigma}_x$, $\dot{\sigma}_y$, and $\dot{\theta}_o$ are zero , and thus V is zero at the equilibrium point. Also, when k is even, *i.e.* $k = 2k'$ for $k' \in \mathbb{Z}$, we have $\theta_o = \theta_o(0) + 2k'\pi$, which results in $\sin(\theta_o) = \sin(\theta_o(0))$ and $\cos(\theta_o) = \cos(\theta_o(0))$. This also leads to zero values for η_{is} and η_{ic} in Eq. 2.100, and consequently zero values for σ_x and σ_y with regards to Eq. 2.102. Therefore, W is zero at this equilibrium point. Furthermore, when k

is odd, *i.e.* $k = 2k' + 1$ for $k' \in \mathbb{Z}$, we have $\theta_o = \theta_o(0) + (2k' + 1)\pi$, which results in $\sin(\theta_o) = -\sin(\theta_o(0))$ and $\cos(\theta_o) = -\cos(\theta_o(0))$. These turn nonzero values for η_{is} and η_{ic}, and consequently, for σ_x and σ_y, which do not necessarily cancel out each other in the summation term in Eq. 2.105. Hence, W is not zero at the equilibrium point with odd k's. □

Theorem 2.2.6. *The trajectories of the system in Eq. 2.57 with the decentralized controllers in Eq. 2.95 are asymptotically stable to the equilibrium point characterized by* $\sigma_x, \sigma_y, \dot{\sigma}_x, \dot{\sigma}_y, \dot{\theta}_o = 0$, *and* $\theta_o = \theta_o(0) + 2k\pi$ *for* $k \in \mathbb{Z}$.

Proof. The time derivative of the Lyapunove function in Eq. 2.105 is calculated as:

$$\begin{aligned}\dot{W} = &- c_t(\dot{\sigma}_x^2 + \dot{\sigma}_y^2) - c_r\dot{\theta}_o^2 + 2k(f_s(\theta_o)\dot{\sigma}_x - f_c(\theta_o)\dot{\sigma}_y)\dot{\theta}_o - k_I N(\sigma_x\dot{\sigma}_x + \sigma_y\dot{\sigma}_y) \\ &+ k_I(\dot{\sigma}_x\eta_s + \sigma_x f_s(\theta_o)\dot{\theta}_o - \dot{\sigma}_y\eta_c - \sigma_y f_c(\theta_o)\dot{\theta}_o) - k_I\sum_{i=1}^{N}(\eta_{i,s}\dot{\eta}_{i,s} + \eta_{i,c}\dot{\eta}_{i,c}) \\ &+ k_I\sum_{i=1}^{N}\left((\sigma_x - \eta_{i,s})(\dot{\sigma}_x - \dot{\eta}_{i,s}) + (\sigma_y + \eta_{i,c})(\dot{\sigma}_y + \dot{\eta}_{i,c})\right). \qquad (2.106)\end{aligned}$$

Using Eq. 2.98 and the fact that $\dot{\eta}_s = f_s(\theta_o)\dot{\theta}_o$ and $\dot{\eta}_c = f_c(\theta_o)\dot{\theta}_o$, we can cancel many terms in the expression for $\dot{W}$ and simplify it to:

$$\dot{W} = -\boldsymbol{z}_1^T\boldsymbol{\Lambda}\boldsymbol{z}_1, \qquad (2.107)$$

where $\boldsymbol{z}_1$ and $\boldsymbol{\Lambda}$ are the same vector and matrix as in Eq. 2.66, respectively. This result at lest shows that W remains bounded as $t \to \infty$. Furthermore, we see that when $\dot{W}$ is identically zero, *i.e.* $\dot{W} \equiv 0$, we have $\dot{\sigma}_x, \dot{\sigma}_y, \dot{\theta}_o \equiv 0$, and taking into account the closed-loop dynamics in Eq. 2.96, we can obtain $\sigma_x, \sigma_y \equiv 0$. Thus, by LaSalle's invariance principle (Khalil, 1996), and invoking *Proposition* 2.2.5, we conclude that the system trajectories asymptotically converge to the aforementioned equilibrium point. In other words, while the objectives with the proportional controller are still achieved, the drift from the desired path is driven to zero. □

2.2.5 Simulation Results

Proportional control

In this section, we validate our analysis with simulation results for collective transport by a team of robots that are arranged in three different distributions around a payload. We study the effect of the robot distribution on the convergence rate of the system to the target transport velocity, the amount of rotation exhibited by the load, and the translational drift of the load from the desired path. The load is modeled as a homogeneous circular ring with mass $m_o = 1$ kg and moment of inertia $I_o = 0.33$ kg·m^2. Six point-mass robots, each with mass $m_r = 0.05$ kg, are rigidly attached to the load. The controller gain is $k = 0.08$ and the target transport speed is $v_{des} = 0.1$ m/s. The simulations were each run for 200 s.

Fig. 2.13-Fig. 2.15 show snapshots of the load over time for each robot distribution. The robot locations are marked as colored points on the perimeter of the load in its initial and final configurations. The target path for the load's center of mass is shown as a red dotted line, and its actual trajectory is plotted in blue. The red line on the load indicates its orientation. In addition, Fig. 2.16 plots the corresponding time evolution of the load's rotation and angular velocity, along with the drift d of the system's center of mass from the target path for all three distributions.

In the first simulation (Fig. 2.13), the robots have an equally-spaced distribution, and the load is transported to the goal with no change in θ_o and no drift d from the target path, as shown in Fig. 2.16. This is because both $||\boldsymbol{r}_c||$ and ξ are zero. For this case, $\epsilon = 0.1584$. In the second simulation (Fig. 2.14), the robots have a

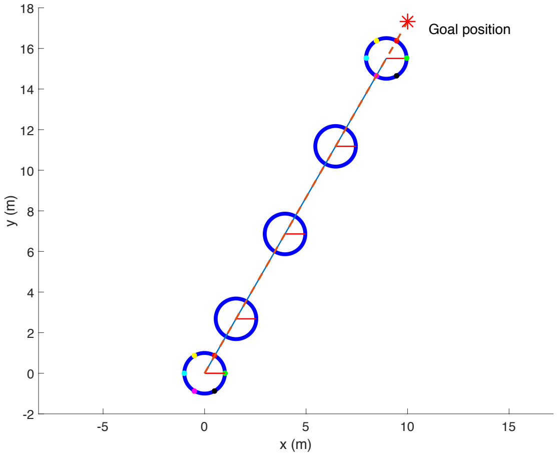


Figure 2.13: Snapshots of the Payload Over Time with an Equally-Spaced Distribution of Robots Around its Perimeter (*Distribution 1*).

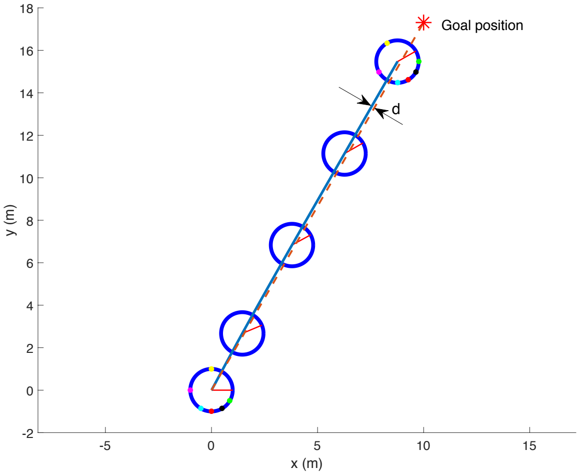


Figure 2.14: Snapshots of the Payload Over Time with a Nonuniform Distribution of Robots Around its Perimeter (*Distribution 2*).

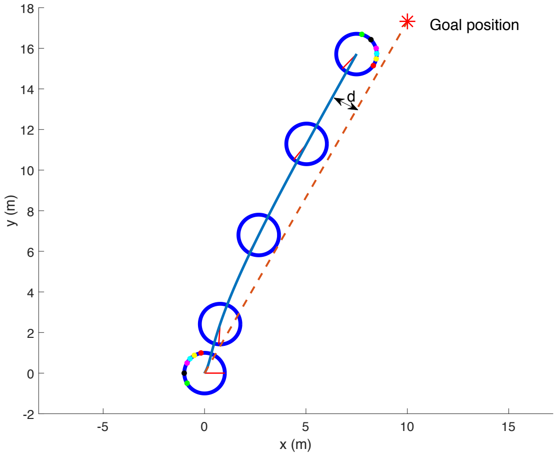


Figure 2.15: Snapshots of the Payload Over Time with a Highly Nonuniform Distribution of Robots Around its Perimeter (*Distribution 3*).

nonuniform distribution for which $\xi = 0.179$ and $||\boldsymbol{r}_c|| = 0.02$. The load undergoes a total rotation of about $\theta_o = 30°$, and its drift from the target path increases to about $d = 15$ cm. For this case, ϵ has decreased slightly to 0.1582. In the third simulation (Fig. 2.15), the robots are clustered within a quarter of the load's perimeter. The load undergoes a large rotation of about $\theta_o = 140°$, and its drift from the target path reaches a maximum of about $d = 1.4$ m. For this case, $||\boldsymbol{r}_c|| = 0.05$ and ξ has increased to 0.3875, which has lowered ϵ to 0.1577.

Finally, Fig. 2.17 shows the time evolution of the variables $s_x \equiv \dot{x}_o - v_{des}$ and $s_y \equiv \dot{y}_o$, the discrepancies between the actual and target velocity components of the system's center of mass, for all three distributions. In all cases, s_x converges to zero at an exponential rate, which is slowest for the third distribution. For the second and third distributions, s_y displays an overshoot before converging to zero, with a much higher overshoot for the third distribution because of its relatively large value of ξ compared to the other two cases. While the second distribution results in convergence to the desired velocity within about 150 s, the third distribution requires more than 200 s to converge.

Proportional-Integral control

The effect of adding the integral control for the third distribution, which had the highest drift, is shown in Fig. 2.18. The system parameters are the same as in the case with proportional control only, and the controller gains are chosen as $k = 0.1$ and $k_I = 0.005$. Fig. 2.18 confirms that the large drift in Fig. 2.15 is driven to zero, and the payload motion converges to the desired path after a transient phase. The convergence of the states of the system in Eq. 2.96 is shown in Fig. 2.19.

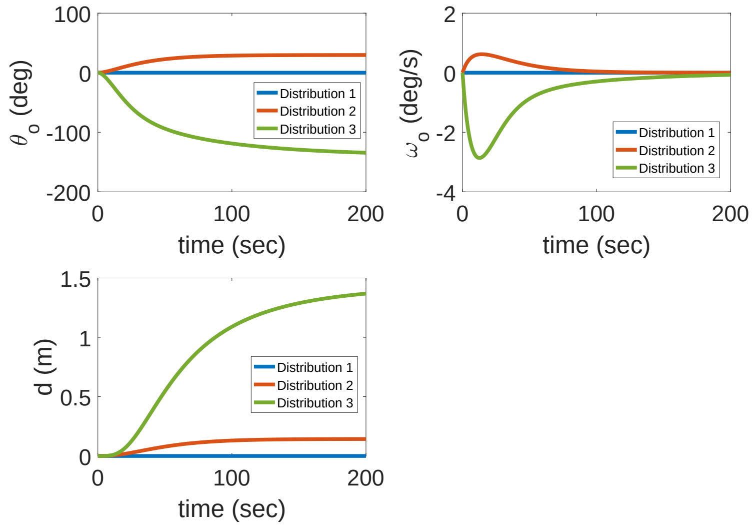


Figure 2.16: Time Evolution of the Load's Rotation θ_o and Angular Velocity $\omega_o = \dot{\theta}_o$ and the Drift d of the System's Center of Mass from the Desired Path for All Three Distributions.

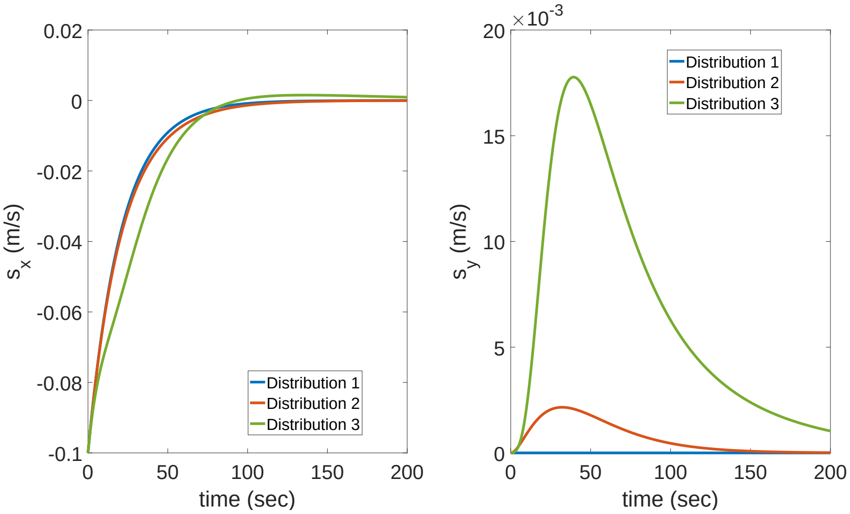


Figure 2.17: Time Evolution of $s_x \equiv \dot{x}_o - v_{des}$ and $s_y \equiv \dot{y}_o$ for All Three Distributions.

2.3 Decentralized Adaptive Controllers for Differential-Drive Mobile Manipulators

This section is an extension to the previous works in Section 2.1 and Section 2.2, in which we considered collective transport scenarios that are similar to those in (Wilson *et al.*, 2016) and (Rubenstein *et al.*, 2013). In those works, we designed decentralized controllers for teams of point-mass robots that are rigidly attached to the payload.

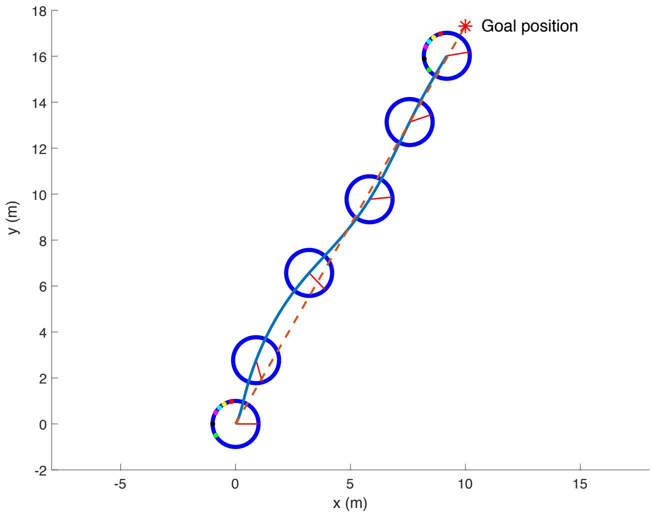


Figure 2.18: Snapshots of the Payload over Time with Proportional-Integral Control for *Distribution 3*.

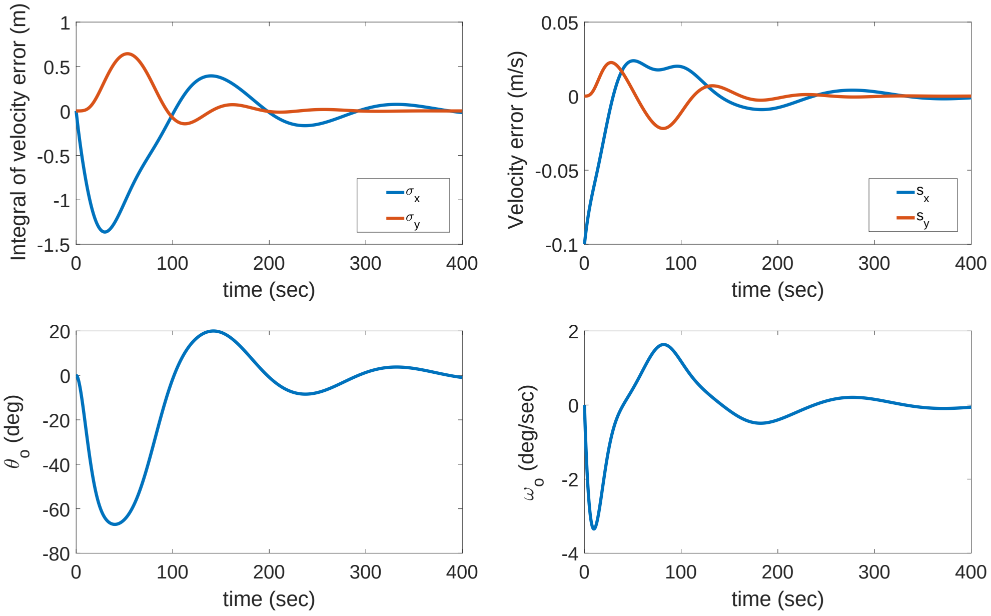


Figure 2.19: Time Evolution of the Integral of Velocity Error (σ_x) and $(\sigma_y = d)$, Velocity Error $(\dot{\sigma}_x = s_x)$ and $(\dot{\sigma}_y = s_y)$, and the Load's Rotation θ_o and Angular Velocity $\omega_o = \dot{\theta}_o$ for *Distribution 3*.

The controllers only required robots' measurements of their own speed and heading, without any information about the number of robots, the payload dynamics, and the robots' distribution around the payload. The controllers drove the robot team to transport the payload toward a target direction with a regulated velocity. Here, we consider a more realistic scenario in which the collective transport task is performed by differential-drive robots with 1-DOF manipulator arms, like the robots in Fig. 1.1.

To derive decentralized controllers that can be implemented on such robots, we make significant modifications to our previous controller design procedure in Section 2.1 and Section 2.2. First, we design the desired manifolds of system motion such that the system trajectories do not violate the holonomic constraints between the robots and the payload. For the case of point-mass robots in (**Farivarnejad, H.** and Berman, 2018), we implicitly incorporated these constraints when we used the kinematics of the payload to derive the dynamics of the system. In this scenario, however, since the robots and their manipulator arms may in general have different initial configurations (see Fig. 2.20), we must explicitly account for the holonomic constraints between the robots and the payload in the design of the desired manifolds of motion. Second, we design adaptive robot controllers that stabilize the system to the desired manifolds of motion and estimate uncertain mass and geometric parameters of the robots. We prove that the closed-loop system will converge to the target transport speed and direction for all initial conditions that are far enough from singular configurations (described in Section 2.3.3). Although adaptive control has previously been used for cooperative manipulation in (Yagiz *et al.*, 2010; Yim *et al.*, 1999; Parra-Vega *et al.*, 2013; Ponce-Hinestroza *et al.*, 2016), these strategies require predefined trajectories for each robot and/or for the payload, as well as knowledge about the robots' distribution around the payload. In addition, the adaptive control strategy in (Lee *et al.*, 2017) requires that the robots have information about the

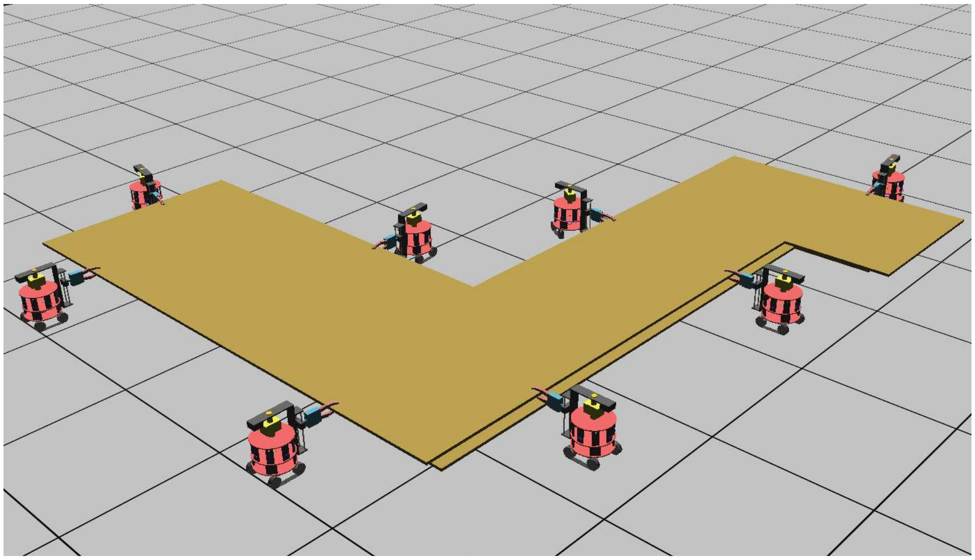


Figure 2.20: Simulated Pheeno Robots (Wilson *et al.*, 2016) Performing a Collective Transport Task.

payload's dynamics. In contrast, the control strategy proposed here only requires local robot measurements of their own motion, and it does not rely on information about the environment, load, or transport team.

2.3.1 Problem Statement

We consider a team of N identical autonomous ground robots that are arranged on a planar surface in an arbitrary configuration around a payload, as in Fig. 2.20. Each robot is comprised of a differential-drive core module, equipped with a 1-DOF manipulator arm that can rotate about the core's central axis. We assume that the manipulator arm of each robot is attached to the load via a point grasp and that the load is lifted above the ground. We also assume that each robot can measure its own speed and heading, as well as the rotation angle and angular velocity of its manipulator. The robots do not have global localization or communication capabilities, and they lack information about the payload's physical properties, velocity, and the position of its center of mass; the number of robots in the transport team; and the robots' distribution around the payload.

Our objective is to design decentralized robot controllers that drive the team to collectively transport the load at a desired speed along a straight path in a target direction. We assume that each robot knows the target speed and direction, although they are not assigned predefined trajectories. To enable the robots to act autonomously during transport, the controllers must not depend on global feedback, which would require the presence of a central supervisor. Instead, the controllers must rely only on the minimal local information that is available to each robot.

2.3.2 Dynamical Model

To derive the equation of motion of each 3-DOF robot in the transport team, we must first choose a vector of generalized coordinates that describe the configuration of the robot, illustrated in Fig. 2.21. We define $\boldsymbol{x}_i = [x_i \; y_i]^T \in \mathbb{R}^2$ as the position of the center of the i^{th} robot's core in the global coordinate frame I, θ_i as its heading angle with respect to the global frame, θ_{R_i} and θ_{L_i} as the angular positions of the right and left wheels, and ϕ_i as the angular position of the manipulator with respect to a coordinate frame R that is fixed to the core. If we select $\boldsymbol{Q}_i = [x_i \; y_i \; \theta_i \; \theta_{R_i} \; \theta_{L_i} \; \phi_i]^T \in \mathbb{R}^6$ as the generalized coordinates for robot i, the dynamics of the robot must include the Lagrange multipliers that are associated with the nonholonomic constraints between the robot's translational and rotational motion. Alternatively, we can use the generalized coordinates $\boldsymbol{q}_i^* = [\theta_{R_i} \; \theta_{L_i} \; \phi_i]^T$ to formulate the dynamics of the robot in an unconstrained form, as described in Appendix A. Moreover, if we use the invertible transformation $\boldsymbol{q}_i = \boldsymbol{T}\boldsymbol{q}_i^*$ with

$$\boldsymbol{T} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} & 0 \\ \frac{r}{b} & \frac{-r}{b} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.108}$$

in which r is the radius of the robot's wheels and b is the distance between the wheels, then we obtain a new vector of generalized coordinates that are more suitable for our control objectives:

$$\boldsymbol{q}_i = \begin{bmatrix} \frac{r}{2}(\theta_{R_i} + \theta_{L_i}) \\ \frac{r}{b}(\theta_{R_i} - \theta_{L_i}) \\ \phi_i \end{bmatrix}. \tag{2.109}$$

Defining ξ_i as the length of the path traveled by the center of the i^{th} robot's core, the time derivative $\dot{\xi}_i$ is the speed of this point. From the kinematics equations for a differential-drive robot, given by Eq. A.5–Eq. A.6 in Appendix A, we find that the time derivative of the first and second elements of $\boldsymbol{q}_i$ are equal to $\dot{\xi}_i$ and $\dot{\theta}_i$, respectively. Therefore,

$$\dot{\boldsymbol{q}}_i = \begin{bmatrix} \dot{\xi}_i & \dot{\theta}_i & \dot{\phi}_i \end{bmatrix}^T. \tag{2.110}$$

These coordinates express the motion of the robot directly in terms of the parameters that we need to control: the robot's translational motion and heading, and its manipulator arm's angular position.

While engaged in cooperative transport, the dynamics of robot i can be written in the following general form (Murray *et al.*, 1994):

$$\boldsymbol{M}_i(\boldsymbol{q}_i)\ddot{\boldsymbol{q}}_i + \boldsymbol{C}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i)\dot{\boldsymbol{q}}_i + \boldsymbol{N}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i) = \boldsymbol{\tau}_i - \boldsymbol{J}_i^T \boldsymbol{F}_i, \tag{2.111}$$

where $\boldsymbol{q}_i \in \mathbb{R}^3$ is the vector of generalized coordinates defined in Eq. 2.109, $\boldsymbol{\tau}_i \in \mathbb{R}^3$ is the vector of actuator torques, $\boldsymbol{F}_i \in \mathbb{R}^2$ is the force exerted on the robot by the payload, $\boldsymbol{J}_i \in \mathbb{R}^{2\times3}$ is the Jacobian matrix of the end-effector's position, $\boldsymbol{M}_i \in \mathbb{R}^{3\times3}$ is the inertia matrix, $\boldsymbol{C}_i \in \mathbb{R}^3$ is the Coriolis matrix, and $\boldsymbol{N}_i \in \mathbb{R}^3$ is a vector that includes the effect of gravity and frictional forces at the joints.

In addition, we derive the the dynamics of the payload during cooperative transport. Let m_o be the mass of the load and J_o be the load's moment of inertia about

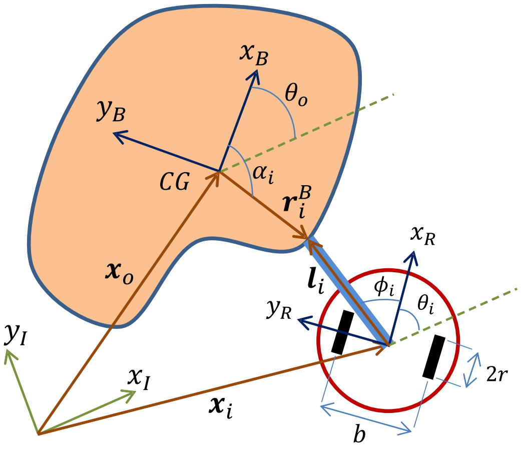


Figure 2.21: A Pheeno Robot in a Collective Transport Task with the Kinematic Chain Representing the Holonomic Constraint Between it and the Payload.

the axis normal to the plane of the motion and passing through its center of gravity (CG). Then $\boldsymbol{M}_o = diag(m_o, m_o, J_o) \in \mathbb{R}^{3\times 3}$ denotes the payload's inertia matrix. We define $\boldsymbol{q}_o = [x_o \; y_o \; \theta_o]^T \in \mathbb{R}^3$ as the payload's vector of generalized coordinates, where x_o and y_o are the position coordinates of the load's CG and θ_o is the load's heading, all with respect to the global frame I. Because there is a point grasp at each robot's attachment point to the load, there exists a kinematic chain passing through the load's CG, the attachment point, and the robot core's center, as illustrated in Fig. 2.21. Then we will use the *grasp matrix* Erhart and Hirche (2016), $\boldsymbol{G} \in \mathbb{R}^{3\times 2N}$, given by:

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{G}_1 & \cdots & \boldsymbol{G}_N \end{bmatrix}, \tag{2.112}$$

where

$$\boldsymbol{G}_i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -||\boldsymbol{r}_i^B|| \sin(\theta_o + \alpha_i) & ||\boldsymbol{r}_i^B|| \cos(\theta_o + \alpha_i) \end{bmatrix}, \tag{2.113}$$

in which $\boldsymbol{r}_i^B \in \mathbb{R}^2$ is the vector from the load's CG to the attachment point of robot i, and α_i is the angle of this vector with respect to the load's local coordinate system. Finally, we define $\boldsymbol{F}_i \in \mathbb{R}^2$ as the force exerted by robot i on the load, expressed in the global frame, and $\boldsymbol{F} \in \mathbb{R}^{2N}$ as the concatenation of all the robots' applied forces:

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{F}_1^T & \cdots & \boldsymbol{F}_N^T \end{bmatrix}^T. \tag{2.114}$$

Then, the dynamics of the payload can be written as

$$\boldsymbol{M}_o \ddot{\boldsymbol{q}}_o = \boldsymbol{G}\boldsymbol{F}. \tag{2.115}$$

2.3.3 *Holonomic Constraints and Design of the Desired Manifolds of Motion*

Using the notation in Fig. 2.21, the kinematic chain that represents the holonomic constraint between a robot and the payload can be expressed as:

$$\boldsymbol{x}_i + \boldsymbol{l}_i - \boldsymbol{R}_B^I \boldsymbol{r}_i^B - \boldsymbol{x}_o \ = \ \boldsymbol{0}, \tag{2.116}$$

in which $\boldsymbol{x}_i = [x_i \ y_i]^T \in \mathbb{R}^2$ is the position of the center of robot i's core in the global frame, $\boldsymbol{l}_i \in \mathbb{R}^2$ is the vector from $\boldsymbol{x}_i$ to the attachment point of robot i on the load, $\boldsymbol{R}_B^I \in \mathbb{R}^{2\times 2}$ is the rotation matrix from the payload's local frame to the global frame, and $\boldsymbol{x}_o = [x_o \ y_o]^T \in \mathbb{R}^2$ is the position of the payload's CG in the global frame. Taking the time derivative of this equation, we can write it in the form of an integrable Pfaffian constraint (Murray *et al.*, 1994),

$$\boldsymbol{A}_i(\boldsymbol{q}_i, \boldsymbol{q}_o) \begin{bmatrix} \dot{\boldsymbol{q}}_i \\ \dot{\boldsymbol{q}}_o \end{bmatrix} = \boldsymbol{0}, \tag{2.117}$$

where $\boldsymbol{A}_i = [\boldsymbol{J}_i \ \ -\boldsymbol{G}_i^T]$, in which

$$\boldsymbol{J}_i = \begin{bmatrix} \cos(\theta_i) & -l_i \sin(\theta_i + \phi_i) & -l_i \sin(\theta_i + \phi_i) \\ \sin(\theta_i) & l_i \cos(\theta_i + \phi_i) & l_i \cos(\theta_i + \phi_i) \end{bmatrix} \tag{2.118}$$

with $l_i = \|\boldsymbol{l}_i\|$, and $\boldsymbol{G}_i$ given by Eq. 2.113.

In the case of N robots, the constraint Eq. 2.116 exists between each robot and the payload. Then Eq. 2.117 can be expanded to include all N constraints:

$$\boldsymbol{A}(\boldsymbol{q}_a)\dot{\boldsymbol{q}}_a = \boldsymbol{0}, \tag{2.119}$$

where $\boldsymbol{q}_a = [\boldsymbol{q}^T \ \boldsymbol{q}_o{}^T]^T \in \mathbb{R}^{3N+3}$, in which $\boldsymbol{q} = [\boldsymbol{q}_1{}^T \ \cdots \ \boldsymbol{q}_N{}^T]^T \in \mathbb{R}^{3N}$, and the *constraint matrix* $\boldsymbol{A} \in \mathbb{R}^{2N\times(3N+3)}$ is given by:

$$\boldsymbol{A}(\boldsymbol{q}_a) = \begin{bmatrix} \boldsymbol{J}_1 & \mathbf{0}_{2\times3} & \cdots & \mathbf{0}_{2\times3} & -\boldsymbol{G}_1^T \\ \mathbf{0}_{2\times3} & \boldsymbol{J}_2 & \mathbf{0}_{2\times3} & \cdots & -\boldsymbol{G}_2^T \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_{2\times3} & \cdots & \mathbf{0}_{2\times3} & \boldsymbol{J}_N & -\boldsymbol{G}_N^T \end{bmatrix}. \tag{2.120}$$

Eq. 2.119 describes the allowable velocities of the system in the entire configuration space. This means that these velocities can evolve only in the null space of $\boldsymbol{A}$, and we must take this fact into account when designing the desired manifolds of motion. Otherwise, the desired system behavior, which is described by the manifolds, would not be reachable by trajectories of the system.

We aim to design robot controllers that do not require any information about the payload's position and velocity and the distribution of robots around the payload. Toward this end, we design the desired manifold of motion for a single robot i by considering its constraint with the payload, defined by the matrix $\boldsymbol{A}_i$, and then showing that the manifolds for all N robots are compatible with the entire set of N constraints, defined by the matrix $\boldsymbol{A}$.

For mechanical systems, any first-order desired manifold of motion can be written in the general form $\boldsymbol{\psi} = \dot{\boldsymbol{\eta}} - \dot{\boldsymbol{\eta}}_r$, where $\dot{\boldsymbol{\eta}}$ is the vector of system velocities and $\dot{\boldsymbol{\eta}}_r$ is the vector of reference velocities, which can be a function of time and $\boldsymbol{\eta}$ (Slotine and Li, 1991). In general, $\boldsymbol{\psi}$ must be driven to zero, at which point $\dot{\boldsymbol{\eta}}$ will track $\dot{\boldsymbol{\eta}}_r$. Here, defining $\boldsymbol{\eta}_i := [\boldsymbol{q}_i^T \ \boldsymbol{q}_o^T]^T \in \mathbb{R}^6$ as the vector of the generalized coordinates of robot i and the payload, we specify a reference velocity vector $\dot{\boldsymbol{\eta}}_{r_i} := [\dot{\boldsymbol{q}}_{r_i}^T \ \dot{\boldsymbol{q}}_{r_o}^T]^T \in \mathbb{R}^6$ that lies in the null space of $\boldsymbol{A}_i$ so that it is achievable by $\dot{\boldsymbol{\eta}}_i$. The null space of $\boldsymbol{A}_i$ can be written as:

$$\mathcal{N}(\boldsymbol{A}_i) = span(\boldsymbol{e}_{1_i}, \ \boldsymbol{e}_{2_i}, \ \boldsymbol{e}_{3_i}, \ \boldsymbol{e}_{4_i}), \tag{2.121}$$

where

$$\begin{aligned}
\boldsymbol{e}_{1_i} &= [\, 0 \ -1 \ 1 \ 0 \ 0 \ 0 \,]^T \ , \\
\boldsymbol{e}_{2_i} &= \left[\ \frac{\cos(\theta_i + \phi_i)}{\cos(\phi_i)} \ \ \frac{-\sin(\theta_i)}{l_i \cos(\phi_i)} \ \ 0 \ 1 \ 0 \ 0 \ \right]^T \ , \\
\boldsymbol{e}_{3_i} &= \left[\ \frac{\sin(\theta_i + \phi_i)}{\cos(\phi_i)} \ \ \frac{\cos(\theta_i)}{l_i \cos(\phi_i)} \ \ 0 \ 0 \ 1 \ 0 \ \right]^T \ , \\
\boldsymbol{e}_{4_i} &= [\, e_{41_i} \ e_{42_i} \ 0 \ 0 \ 0 \ 1 \,]^T \ , \qquad (2.122)
\end{aligned}$$

in which the first two elements of the vector $\boldsymbol{e}_{4_i}$ are:

$$\begin{aligned}
e_{41_i} &= \frac{-||\boldsymbol{r}_i^B|| \sin(\theta_o + \alpha_i - \theta_i - \phi_i)}{\cos(\phi_i)}, \\
e_{42_i} &= \frac{||\boldsymbol{r}_i^B|| \cos(\theta_o + \alpha_i - \theta_i - \phi_i)}{\cos(\phi_i)}. \qquad (2.123)
\end{aligned}$$

Indeed, the vector $\dot{\boldsymbol{\eta}}_{r_i}$ must be a linear combination of these four vectors, since they span $\mathcal{N}(\boldsymbol{A}_i)$. As stated in section 2.3.1, the desired motion for the payload is a regulated speed along the x_I direction, zero speed along the y_I direction, and zero angular speed. Thus, we set:

$$\dot{\boldsymbol{q}}_{r_o} = [v_{des} \ 0 \ 0]^T, \qquad (2.124)$$

where v_{des} is the desired speed of transport. Then, noting the 1's in the fifth and sixth elements of $\boldsymbol{e}_{3_i}$ and $\boldsymbol{e}_{4_i}$, respectively, we can conclude that $\dot{\boldsymbol{\eta}}_{r_i}$ cannot have any projection on these two vectors and must be a linear combination of only $\boldsymbol{e}_{1_i}$ and $\boldsymbol{e}_{2_i}$. Moreover, the desired motion for the robot includes a regulated forward speed and a zero heading angle, in addition to a zero angular speed for the manipulator.

Here, we design the desired manifolds in a way that is applicable to general initial configurations of robots, in which the robots have an arbitrary distribution around the payload and arbitrary headings and manipulator angles, as depicted in Fig. 1.1. However, we must first take into account the fact that both $\boldsymbol{e}_{2_i}$ and $\boldsymbol{e}_{3_i}$ have entries with $\cos(\phi_i)$ in the denominator, and hence $\phi_i = k\pi/2$ for any integer k is a singular configuration that has to be avoided. Therefore, we consider the following assumption, which can be enforced by restricting the range of rotation of the manipulator arm (for example, $\phi_i \in 0.9[-\pi/2, \pi/2]$):

Assumption 2.3.1. *Each robot starts the transport from a configuration far from singular configurations, and stays far from these configurations during the entire transport.*

The reference velocity vector is specified as

$$\dot{\boldsymbol{\eta}}_{r_i} \;=\; v_{des}\boldsymbol{e}_{2_i}, \tag{2.125}$$

and therefore, the desired manifold of motion is given by

$$\boldsymbol{\psi}_i = \dot{\boldsymbol{\eta}}_i - \dot{\boldsymbol{\eta}}_{r_i}. \tag{2.126}$$

This manifold can be written as $\boldsymbol{\psi}_i = [\boldsymbol{s}_i^T \; \boldsymbol{s}_o^T]^T$, where $\boldsymbol{s}_i$ and $\boldsymbol{s}_o$ are the desired manifolds for robot i and the payload, respectively:

$$\boldsymbol{s}_i = \begin{bmatrix} \dot{\xi}_i - v_{des}\frac{\cos(\theta_i+\phi_i)}{\cos(\phi_i)} \\ \dot{\theta}_i + v_{des}\frac{\sin(\theta_i)}{l_i\cos(\phi_i)} \\ \dot{\phi}_i \end{bmatrix}, \; \boldsymbol{s}_o = \begin{bmatrix} \dot{x}_o - v_{des} \\ \dot{y}_o \\ \dot{\theta}_o \end{bmatrix}. \tag{2.127}$$

We see that $\boldsymbol{s}_o$ is independent of the robot's state variables. We now show that the reference velocity vector Eq. 2.125 is compatible with all holonomic constraints in the system, *i.e.* Eq. 2.119, and then we prove that it produces the desired motion characteristics of the payload and the robots.

Proposition 2.3.2. *The desired velocities in Eq. 2.125 are reachable by all robots in the team and the payload during collective transport.*

Proof. Since both $\dot{\boldsymbol{\eta}}_i$ and $\dot{\boldsymbol{\eta}}_{r_i}$ are in $\mathcal{N}(\boldsymbol{A}_i)$ for each robot $i = 1, ..., N$, we can conclude that $\boldsymbol{A}_i\boldsymbol{\psi}_i = \boldsymbol{0}$, $i = 1, ..., N$, which implies that

$$\begin{bmatrix} \boldsymbol{J}_i & -\boldsymbol{G}_i^T \end{bmatrix} \begin{bmatrix} \boldsymbol{s}_i \\ \boldsymbol{s}_o \end{bmatrix} = \boldsymbol{0}, \quad i = 1, ..., N. \tag{2.128}$$

Defining $\boldsymbol{s} := [\boldsymbol{s}_1{}^T \ \cdots \ \boldsymbol{s}_N{}^T]^T$ and $\boldsymbol{s}_a := [\boldsymbol{s}^T \ \boldsymbol{s}_o{}^T]^T$, we can rewrite the N equations in Eq. 2.128 in the following compact form:

$$\boldsymbol{A}(\boldsymbol{q}_a)\boldsymbol{s}_a = \boldsymbol{0}, \tag{2.129}$$

which means that $\boldsymbol{s}_a \in \mathcal{N}(\boldsymbol{A})$. Since the designed manifold $\boldsymbol{s}_a$ therefore satisfies the constraint Eq. 2.119, the reference velocity vector $\dot{\boldsymbol{\eta}}_{r_i}$ is reachable by each robot i.

□

Proposition 2.3.3. *On the manifolds $\boldsymbol{\psi}_i$ defined by Eq. 2.126, the payload's motion converges to a pure translation along the x_I direction with speed v_{des}; the robots' speeds and headings converge to v_{des} and 0, respectively; and the robots' manipulators converge to a stationary configuration.*

Proof. At the time when $\boldsymbol{\psi}_i = \boldsymbol{0}$, we have $\dot{\theta}_o = 0$, which means that the payload has stopped rotating, and $\dot{y}_o = 0$, $\dot{x}_o = v_{des}$, which means that the payload is moving along the x_I direction at the desired speed. In addition, from the elements of $\boldsymbol{s}_i$, we have that $\dot{\phi}_i = 0 \ \rightarrow \ \phi_i = const. := \phi_{i_{ss}}$, which means that the manipulator of robot i has stopped rotating. Consequently, we can conclude that $\cos(\phi_i) = const. = \cos(\phi_{i_{ss}})$, and thus the second element of $\boldsymbol{s}_i$, which governs the dynamics of the robot's heading, can be written in the following form:

$$\dot{\theta}_i + c_i \sin(\theta_i) = 0, \tag{2.130}$$

where $c_i = \frac{v_{des}}{l_i \cos(\phi_{i_{ss}})}$ is constant and positive for $\phi_i \in (-\pi/2, \pi/2)$, which is the manipulator's range of motion. We can show that θ_i asymptotically converges to zero as follows. We consider the following Lyapunov function W and its time derivative along the trajectories of Eq. 2.130:

$$W = 1 - \cos(\theta_i) \quad \rightarrow \quad \dot{W} = -c_i \sin(\theta_i)^2. \tag{2.131}$$

Since $\dot{W}$ is negative definite, Eq. 2.130 is asymptotically stable at $\theta_i = 0$, which means that robot i will converge to the desired heading.[1] Finally, since $\boldsymbol{s}_i = \boldsymbol{0}$ and θ_i converges to zero, we can write the following for the first element of $\boldsymbol{s}_i$:

$$\lim_{t \to \infty} \dot{\xi}_i = v_{des} \lim_{\theta_i \to 0} \frac{\cos(\theta_i + \phi_{i_{ss}})}{\cos(\phi_{i_{ss}})} = v_{des}, \tag{2.132}$$

which shows that the robot's speed will converge to the desired value. □

Remark 2.3.4. *The compatibility of the desired manifolds of motion in Eq. 2.126 with the holonomic constraints between the robots and the payload enables the robots to perform the transport task without information about the position of the payload's center of mass or the vector from this point to each robot's attachment point, i.e.* $\boldsymbol{r}_i^B$*. This information is required in many collective transport methods that have been proposed in the literature, as described in Section 2.3.1.*

2.3.4 Controller Design and Stability Analysis

As discussed in section 2.3.3, the desired system behavior is achieved on the manifolds expressed as $\boldsymbol{\psi}_i = [\boldsymbol{s}_i^T \ \boldsymbol{s}_o^T]^T$, in which $\boldsymbol{s}_i := [s_{\xi_i} \ s_{\theta_i} \ s_{\phi_i}]^T$ is associated with the dynamics of robot i, and $\boldsymbol{s}_o$ is associated with the payload's dynamics. Moreover, it is possible to verify that the terms $\boldsymbol{M}_i(\boldsymbol{q}_i)$, $\boldsymbol{C}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i)$, and $\boldsymbol{N}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i)$ in Eq. 2.111 can be linearly parametrized in terms of a constant vector $\boldsymbol{\Theta}_i \in \mathbb{R}^P$ that contains

[1] To keep $\theta_i(t) \in (-\pi, \pi)$, the robot moves backward when the absolute value of its initial heading error is more than $\pi/2$. This is implemented by switching the desired speed to $-v_{des}$ and shifting the desired heading by $-\pi$ rad.

P uncertain mass and geometric properties of robot i (Slotine and Li, 1991). Thus, we can define a matrix $\boldsymbol{Y}_i = \boldsymbol{Y}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i, \dot{\boldsymbol{q}}_{r_i}, \ddot{\boldsymbol{q}}_{r_i}) \in \mathbb{R}^{3\times P}$, which is a function of the reference quantities $\dot{\boldsymbol{q}}_{r_i}, \ddot{\boldsymbol{q}}_{r_i}$ and the measured quantities $\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i$, such that:

$$\boldsymbol{M}_i(\boldsymbol{q}_i)\ddot{\boldsymbol{q}}_{r_i} + \boldsymbol{C}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i)\dot{\boldsymbol{q}}_{r_i} + \boldsymbol{N}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i) = \boldsymbol{Y}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i, \dot{\boldsymbol{q}}_{r_i}, \ddot{\boldsymbol{q}}_{r_i})\boldsymbol{\Theta}_i. \tag{2.133}$$

Here, we assume that we have uncertain estimates of $P = 3$ parameters for each robot: the mass and moment of inertia of its core, and the mass of its manipulator.

We now design the controller for the actuator torque that is applied by the wheels and manipulator of robot i. This torque is defined as $\boldsymbol{\tau}_i^* = \boldsymbol{T}^{-1}\boldsymbol{\tau}_i$, where

$$\boldsymbol{\tau}_i = -\boldsymbol{K}\boldsymbol{s}_i + \boldsymbol{Y}_i\hat{\boldsymbol{\Theta}}_i. \tag{2.134}$$

Here, $\boldsymbol{K} \in \mathbb{R}^{3\times 3}$ is a positive definite matrix that contains the controller gains, and $\hat{\boldsymbol{\Theta}}_i$ is an estimate of $\boldsymbol{\Theta}_i$, which is updated according to the following adaptation law:

$$\dot{\hat{\boldsymbol{\Theta}}}_i = -\boldsymbol{\Gamma}\boldsymbol{Y}_i^T\boldsymbol{s}_i, \tag{2.135}$$

in which $\boldsymbol{\Gamma} \in \mathbb{R}^{3\times 3}$ is a symmetric positive definite matrix that contains the adaptation gains.

Remark 2.3.5. *The controller in Eq. 2.134 and the adaptation law in Eq. 2.135 are completely decentralized, in the sense that each robot i can execute them using only measurements of its own motion (i.e., the quantities in $\boldsymbol{s}_i$ and $\boldsymbol{Y}_i$). They also do not require any information about the payload's motion and the robots' distribution around the payload.*

To ensure that the controller in Eq. 2.134 and the adaptation law in Eq. 2.135 drive the system trajectories to the desired manifolds, we prove the stability of the closed-loop system in Theorem 2.3.7. First, we state the following lemma, which will be used in the proof of the theorem.

Lemma 2.3.6. *Consider a team of $N \geq 2$ robots that are attached to a payload at distinct points, with the robots' reference velocities specified as in Eq. 2.125. If $\boldsymbol{s}_i \equiv \boldsymbol{0}$ for each robot $i \in \{1, 2, ..., N\}$, then $\boldsymbol{s}_o \equiv \boldsymbol{0}$.*

Proof. Since both $\dot{\boldsymbol{q}}_a$ and $\dot{\boldsymbol{q}}_{a_r}$ are in $\mathcal{N}(\boldsymbol{A})$, we can write $\boldsymbol{A}\boldsymbol{s}_a = \boldsymbol{0}$. Thus, according to Eq. 2.119, we have:

$$\boldsymbol{J}\boldsymbol{s} = \boldsymbol{G}^T \boldsymbol{s}_o, \tag{2.136}$$

in which $\boldsymbol{J} \in \mathbb{R}^{2N \times 3N}$ is a rectangular matrix in block-diagonal form, with the blocks defined as $\boldsymbol{J}_i$, $i = 1, ..., N$. When $\boldsymbol{s}_i = \boldsymbol{0}$ for all $i = 1, ..., N$, then $\boldsymbol{s} = \boldsymbol{0}$. This implies that $\boldsymbol{J}\boldsymbol{s} = \boldsymbol{0}$, which means that $\boldsymbol{G}^T \boldsymbol{s}_o = \boldsymbol{0}$. Moreover, $\boldsymbol{G} \in \mathbb{R}^{3 \times 2N}$, and its first two rows, which are the first two columns of $\boldsymbol{G}^T$, are linearly independent. The third row is linearly dependent on the other rows only in the case where $||\boldsymbol{r}_i^B|| \sin(\theta_o + \alpha_i) = ||\boldsymbol{r}_j^B|| \sin(\theta_o + \alpha_j)$ and $||\boldsymbol{r}_i^B|| \cos(\theta_o + \alpha_i) = ||\boldsymbol{r}_j^B|| \cos(\theta_o + \alpha_j)$ $\forall\ i, j \in \{1, 2, ..., N\}$, which is impossible because the robots are attached to distinct points on the payload. Therefore, all columns of $\boldsymbol{G}^T$ are linearly independent, and so $rank(\boldsymbol{G}^T) = 3$, which implies that $dim(\boldsymbol{\mathcal{N}}(\boldsymbol{G}^T)) = 3 - 3 = 0$. Hence, the null space of $\boldsymbol{G}^T$ is empty, and the only solution for $\boldsymbol{G}^T \boldsymbol{s}_o = \boldsymbol{0}$ is $\boldsymbol{s}_o \equiv \boldsymbol{0}$. □

Theorem 2.3.7. *Consider a team of N differential-drive robots, each with a 1-DOF manipulator arm that is attached to a payload via a point grasp, as depicted in Fig. 1.1. Given the manifold in Eq. 2.127 and the controller and adaptation law in Eq. 2.134 and Eq. 2.135, the entire system converges to the desired motion, which is defined as the translation of the payload and robots in a specified direction at a target speed without rotation, with the robots' manipulators fixed in a stationary configuration.*

Proof. We consider the following Lyapunov function (Slotine and Li, 1991):

$$V = \frac{1}{2}\sum_{i=1}^{N} \boldsymbol{s}_i^T \boldsymbol{M}_i \boldsymbol{s}_i + \frac{1}{2}\boldsymbol{s}_o^T \boldsymbol{M}_o \boldsymbol{s}_o + \frac{1}{2}\sum_{i=1}^{N} \tilde{\boldsymbol{\Theta}}_i^T \boldsymbol{\Gamma}^{-1} \tilde{\boldsymbol{\Theta}}_i, \tag{2.137}$$

in which $\tilde{\boldsymbol{\Theta}}_i = \hat{\boldsymbol{\Theta}}_i - \boldsymbol{\Theta}_i$ is the parameter estimation error. The time derivative of this function is:

$$\dot{V} = \sum_{i=1}^{N} \boldsymbol{s}_i^T \boldsymbol{M}_i \dot{\boldsymbol{s}}_i + \frac{1}{2}\sum_{i=1}^{N} \boldsymbol{s}_i^T \dot{\boldsymbol{M}}_i \boldsymbol{s}_i + \boldsymbol{s}_o^T \boldsymbol{M}_o \dot{\boldsymbol{s}}_o + \frac{1}{2}\boldsymbol{s}_o^T \dot{\boldsymbol{M}}_o \boldsymbol{s}_o + \sum_{i=1}^{N} \dot{\tilde{\boldsymbol{\Theta}}}_i^T \boldsymbol{\Gamma}^{-1} \tilde{\boldsymbol{\Theta}}_i. \tag{2.138}$$

Since $\boldsymbol{M}_o$ is constant, and by Eq. 2.124, $\ddot{\boldsymbol{q}}_{r_o}$, which is included in $\dot{\boldsymbol{s}}_o$, is equal to zero, $\dot{V}$ is reduced to:

$$\dot{V} = \sum_{i=1}^{N} \boldsymbol{s}_i^T \boldsymbol{M}_i (\ddot{\boldsymbol{q}}_i - \ddot{\boldsymbol{q}}_{r_i}) + \frac{1}{2}\sum_{i=1}^{N} \boldsymbol{s}_i^T \dot{\boldsymbol{M}}_i \boldsymbol{s}_i + \boldsymbol{s}_o^T \boldsymbol{M}_o \ddot{\boldsymbol{q}}_o + \sum_{i=1}^{N} \dot{\tilde{\boldsymbol{\Theta}}}_i^T \boldsymbol{\Gamma}^{-1} \tilde{\boldsymbol{\Theta}}_i. \tag{2.139}$$

Furthermore, considering the passivity property of the robot dynamics in Eq. 2.111 (Murray *et al.*, 1994), recognizing that the matrix $\dot{\boldsymbol{M}}_i - 2\boldsymbol{C}_i$ is skew-symmetric, noting the payload's dynamics in Eq. 2.115, and substituting the controller Eq. 2.134 into the expression for $\boldsymbol{M}_i \ddot{\boldsymbol{q}}_i$ from Eq. 2.111, $\dot{V}$ can be rewritten as:

$$\begin{aligned}\dot{V} = &-\sum_{i=1}^{N} \boldsymbol{s}_i^T \boldsymbol{K} \boldsymbol{s}_i + \sum_{i=1}^{N} \boldsymbol{s}_i^T \boldsymbol{Y}_i \hat{\boldsymbol{\Theta}}_i - \sum_{i=1}^{N} \boldsymbol{s}_i^T \left(\boldsymbol{M}_i(\boldsymbol{q}_i)\ddot{\boldsymbol{q}}_{r_i} + \boldsymbol{C}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i)\dot{\boldsymbol{q}}_{r_i} + \boldsymbol{N}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i)\right) \\ &+ \sum_{i=1}^{N} \dot{\tilde{\boldsymbol{\Theta}}}_i^T \boldsymbol{\Gamma}^{-1} \tilde{\boldsymbol{\Theta}}_i - \begin{bmatrix} \boldsymbol{s}^T & \boldsymbol{s}_o^T \end{bmatrix} \begin{bmatrix} \boldsymbol{J}^T \\ -\boldsymbol{G} \end{bmatrix} \boldsymbol{F}. \end{aligned} \tag{2.140}$$

From Eq. 2.120, the last term on the right-hand side of Eq. 2.140 can be rewritten as $(\boldsymbol{A}\boldsymbol{s}_a)^T \boldsymbol{F}$, and invoking Eq. 2.129, we conclude that this term is zero. Finally, applying the linear parameterization in Eq. 2.133 and the adaptation law in Eq. 2.135, and also using the fact that $\dot{\tilde{\boldsymbol{\Theta}}}_i = \dot{\hat{\boldsymbol{\Theta}}}_i$ since $\boldsymbol{\Theta}_i$ is constant, we have:

$$\dot{V} = -\sum_{i=1}^{N} \boldsymbol{s}_i^T \boldsymbol{K} \boldsymbol{s}_i. \tag{2.141}$$

The negative semi-definiteness of $\dot{V}$ implies the global stability of the system and, consequently, the boundedness of $\boldsymbol{s}_o$, $\boldsymbol{s}_i$, and $\tilde{\boldsymbol{\Theta}}_i$ for all $i = 1, ..., N$. From Eq. 2.134, this result implies the boundedness of each $\boldsymbol{\tau}_i$. By eliminating the vector $\boldsymbol{F}$ from the dynamics of the N robots (Eq. 2.111) and the payload (Eq. 2.115), we observe that $\boldsymbol{\tau}_i$, $i = 1, ..., N$, are the only active torques affecting the dynamics of the entire system of the robots and payload. Since these torques are bounded, we can conclude that $\dot{\boldsymbol{q}}_i$ and $\ddot{\boldsymbol{q}}_i$ are bounded for each robot i. The boundedness of $\ddot{\boldsymbol{q}}_{r_i}$ can be verified from Eq. 2.127. Since $\ddot{\boldsymbol{q}}_i$ and $\ddot{\boldsymbol{q}}_{r_i}$ are both bounded, we have that $\dot{\boldsymbol{s}}_i = \ddot{\boldsymbol{q}}_i - \ddot{\boldsymbol{q}}_{r_i}$ is also bounded. Furthermore, the second time derivative of V can be calculated as:

$$\ddot{V} = -2 \sum_{i=1}^{N} \boldsymbol{s}_i^T \boldsymbol{K} \dot{\boldsymbol{s}}_i. \tag{2.142}$$

Given that $\boldsymbol{s}_i$ and $\dot{\boldsymbol{s}}_i$ are bounded for all i as discussed above, this equation indicates that $\ddot{V}$ is bounded as well. By *Barbalat's lemma* (Slotine and Li, 1991), the positive definiteness of V and the boundedness of $\ddot{V}$ imply that $\dot{V} \to 0$, and consequently $\boldsymbol{s}_i \to 0$, as $t \to \infty$. Finally, from Lemma 2.3.6, we conclude that $\boldsymbol{s}_o \to 0$ as $t \to \infty$, which completes the proof. □

Proposition 2.3.8. *All internal forces $\boldsymbol{F}_i$, which are exerted by the robots on the payload, remain bounded during transport.*

Proof. In the proof of Theorem 2.3.7, we showed that $\dot{\boldsymbol{q}}_i$, $\ddot{\boldsymbol{q}}_i$, and $\boldsymbol{\tau}_i$ are all bounded. We can then conclude from the robot equations of motion 2.111 that the term $\boldsymbol{J}_i^T \boldsymbol{F}_i$ is also bounded for each robot i. In addition, we can confirm that the null space of $\boldsymbol{J}_i^T$ is empty. This implies that all internal forces $\boldsymbol{F}_i$ remain bounded as well. □

2.3.5 Simulation Results

We validated our adaptive control strategy with high-fidelity 3D physics simulations in the robot simulator Webots (Michel, 2004). The robots in the Webots simulations are 3D models of a small mobile robot platform, *Pheeno*, that has been developed in our lab (Wilson *et al.*, 2016).

We implemented the controller and adaptation law proposed in section 2.3.4 in a Webots simulation in which eight Pheeno robots transport the payload to a goal that is located at a heading of $\gamma = 30^\circ$ in the inertial frame. The desired load speed is $v_{des} = 0.2$ m/s, and the load mass and moment of inertia are 1 kg and 0.33 kg·m^2, respectively. The matrices of controller and adaptation gains were set to $\boldsymbol{K} = diag(0.002, 0.006, 0.01)$ and $\boldsymbol{\Gamma} = diag(0.3, 0.5, 0.2)$. The system was simulated for 150 s.

Four snapshots of the simulation are shown in Fig. 2.22, in which the goal location is indicated by the green cone and the desired path of the payload's CG is illustrated by the blue line. Fig. 2.23 plots the load and robot trajectories, which are straight and parallel in the desired direction without significant rotation for the load.

Fig. 2.24 displays the time evolution of the entries of the desired manifold $\boldsymbol{s}_i$ for each robot i (here we drop the i subscripts for simplicity): s_ξ, s_θ, and s_ϕ, which are associated with the robot's speed $\dot{\xi}$, heading angle θ, and manipulator angle ϕ, respectively. The figure also plots the time evolution of $\dot{\xi}$, θ, and ϕ for each robot. The variables are only plotted over the beginning of the simulation in order to clearly illustrate their transient dynamics. The plots show that s_ξ, s_θ, and s_ϕ all converge to zero for each robot, and although they initially exhibit oscillations, they have smooth profiles after ~ 6 s. The manipulator angles of the robots all converge to steady-state values, demonstrating that each robot converges to a fixed configuration. Note that these angles remain far from the singular configuration, *i.e.* $\pm 90^\circ$, in accordance with Assumption 2.3.1. Furthermore, four of the robots, which push the payload, converge to a heading of 30° and speed of 0.2 m/s, while the other four robots, which pull the payload, converge to a heading of -150° and speed of -0.2 m/s. This discrepancy

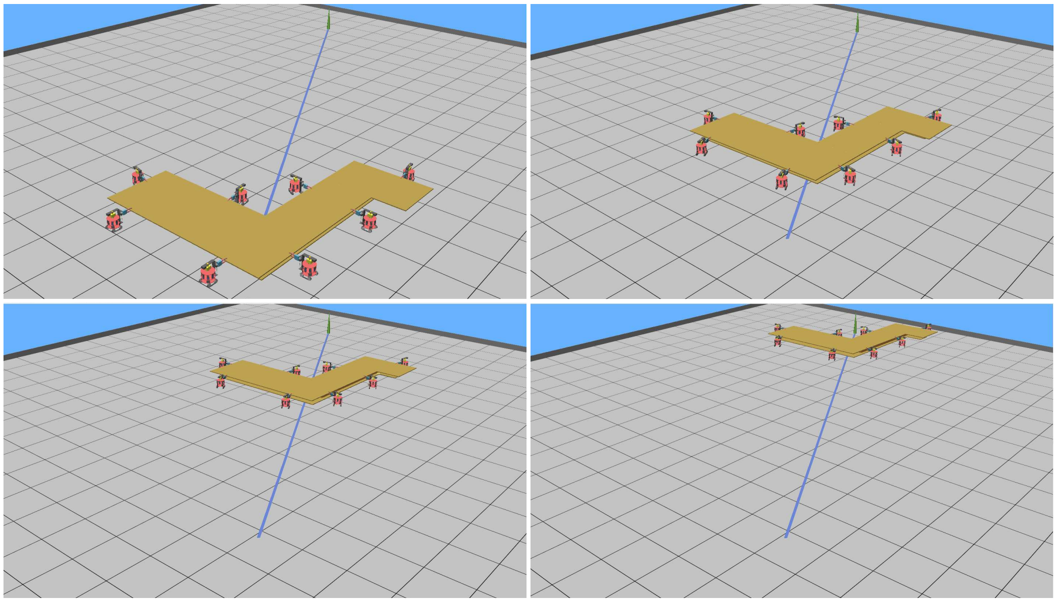


Figure 2.22: Collective Transport by Eight Pheenos Simulated in Webots.

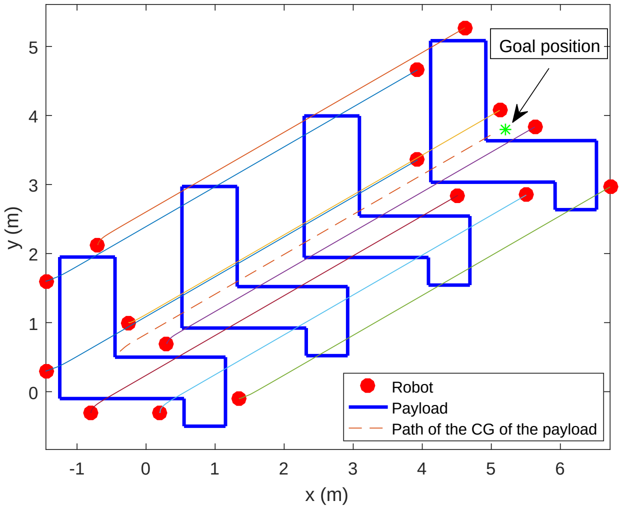


Figure 2.23: The Trajectories of the Pheenos and the Load During Transport.

in heading and speed between the pushing and pulling robots happens due to the maneuver described in the footnote in section 2.3.3. This maneuver prevents the robots from performing unnecessarily large rotations that would slow down their response and possibly drive them to singular configurations.

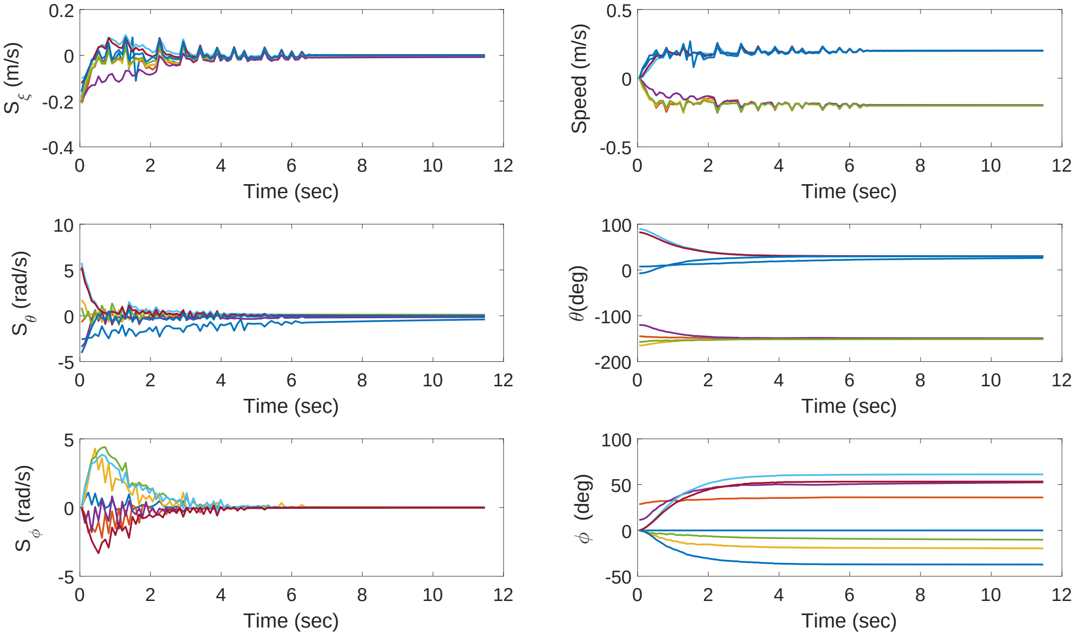


Figure 2.24: Time Evolution of Variables in the Webots Simulation of Collective Transport. *Left Column:* Entries of the Desired Manifold $\boldsymbol{s}_i$ for Each Robot i: s_ξ, s_θ, and s_ϕ (We drop the i subscripts for simplicity). *Right Column:* Speed $\dot{\xi}$, Heading Angle θ, and Manipulator Angle ϕ of Each Robot.

Chapter 3

VELOCITY CONTROL OF COLLECTIVE TRANSPORT IN UNBOUNDED DOMAINS WITH STRICTLY CONVEX OBSTACLES

In this chapter, we propose an obstacle avoidance controller for a disk-shaped holonomic robot with double-integrator dynamics and local sensing. The control objective is for the robot to converge to a target velocity while avoiding collisions with strictly convex obstacles in an unbounded environment. We assume that the robot has no information about the location and geometry of the obstacles, has no localization capabilities, and can only measure its own velocity and its relative position vector to the closest point on any obstacles in its sensing range. We first propose a potential-based controller for the case with a single obstacle, and we prove that the robot safely navigates past the obstacle and attains the desired velocity. For the case with multiple obstacles, we propose a switching control scheme in which the robot applies the single-obstacle controller for the closest obstacle at each instant. We investigate the correctness of this switching control law and demonstrate the absence of local stable equilibrium points that would trap the robot. We validate our analytical results through simulations of a robot that uses the proposed controllers to successfully navigate through an environment with strictly convex obstacles of various shapes and sizes. We finally combine the proposed obstacle avoidance controller with the PI controller in Section 2.2 to design a decentralized control law for velocity control of collective transport in unbounded domains with convex obstacles.

The proposed controller is suitable for applications in which it is necessary to regulate the velocity of a robot and navigate it safely through an unknown, obstacle-filled environment where precise position feedback is absent, unreliable, or not required.

For example, underwater robots may lack accurate global position information via odometry or GPS, only obtaining GPS readings when they surface periodically. A multi-robot control problem that involves velocity regulation and does not require position feedback is flocking control of a group of agents (Tanner *et al.*, 2003), which may need to avoid unanticipated obstacles along their way while stabilizing their velocities and maintaining group cohesion (Sakai *et al.*, 2017).

To summarize, the novel contribution of this work is a robot controller for velocity regulation and obstacle avoidance with all of the following properties:

- The controller does not require that the robot have exact or approximate global position information or *a priori* information about the locations, geometries, or configuration of obstacles in the environment.

- The robot has no predefined trajectory and operates autonomously with minimal capabilities: it can only measure its own velocity and its relative position vector to the closest point on any nearby obstacles within its sensing range.

- The controller has theoretical guarantees on performance; specifically, it can be proved that a robot with this controller will converge to a desired velocity without colliding with obstacles or becoming entrapped by local minima.

3.1 Problem Statement

We consider a disk-shaped holonomic robot that moves in a planar unbounded domain with second-order dynamics (a double-integrator model), $\ddot{\boldsymbol{q}} = \boldsymbol{u}$, where $\boldsymbol{q} = (x, y)^T \in \mathbb{R}^2$ denotes the position of the robot's center in a global reference frame and $\boldsymbol{u} \in \mathbb{R}^2$ is the robot's control input. A physical realization of such a robot is an omnidirectional mobile robot that can move in any direction in a plane at each time instant (Bräunl, 2008). We assume that the domain contains multiple strictly convex obstacles. The control objective is for the robot to attain a desired velocity $\boldsymbol{v}_{des}$ while avoiding collisions with the obstacles. The x-axis of the global reference frame is defined along the direction of $\boldsymbol{v}_{des}$, without loss of generality. We now define two terms that we will frequently use throughout the paper.

Definition 3.1.1. *The line from the robot's current position $\boldsymbol{q}$ that is normal to the obstacle's boundary intersects the boundary at the* ***projection point****. This point and its position vector are denoted by P and $\boldsymbol{q}_P$, respectively, as shown in Fig. 3.1.*

Definition 3.1.2. *The vector $\boldsymbol{q} - \boldsymbol{q}_P$ from the projection point to the robot's current position is called the* ***collision vector****. This vector is denoted by $\mathbf{d}$ and is shown in red in Fig. 3.1.*

We make the following assumptions about the robot's specifications and capabilities. The robot has a circular shape with radius r. It does not have global position information (e.g., GPS) and has no prior knowledge of the obstacles' locations and shapes. The only information provided to the robot is the target velocity $\boldsymbol{v}_{des}$. The robot can measure its own velocity, for example by using tachometers or a velocity estimation algorithm based on optical flow (Ho *et al.*, 2017). It can measure its heading in the global frame, e.g., using a compass.[1] It can also identify the boundaries of nearby obstacles within its local sensing range, which is assumed to be a circle with radius δ_c. We assume that at each time instant, the robot can measure its distance from each obstacle within its sensing range, for example, using infrared sensors or LIDAR. This distance is the length of the collision vector $\boldsymbol{d}$, according to the *Projection Theorem* in (Bertsekas *et al.*, 2003). We also assume that the robot can measure the

[1] Note that the robot's ability to measure its heading (or orientation in general) does not contradict the assumption that it lacks global position information. The orientation of a mobile robot is often measured by its on-board sensors, such as a compass, IMU, or gyroscope, and not necessarily by an external localization system.

angle $\phi_{\boldsymbol{d}}$ of the vector $-\boldsymbol{d}$ in its body-fixed frame, e.g. using LIDAR. By adding $\phi_{\boldsymbol{d}}+\pi$ rad to the robot's heading in the global frame, the robot can obtain the angle of $\boldsymbol{d}$ in the global frame, which we denote by $\theta_{\boldsymbol{d}}$. This angle is required in the proposed control law described in Section 3.2.

Given this minimal and completely local information, we first seek a control law that can solve the following problem.

Problem 3.1.3. *We consider an unbounded domain that contains a single strictly convex obstacle with an arbitrary boundary* [2] *described by* $\beta(x, y) = 0$*, where* $\beta : \mathbb{R}^2 \mapsto \mathbb{R}$ *is at least twice continuously differentiable. We design a robot control law that uses only the local measurements available to the robot to ensure that the robot:*

(1) asymptotically converges to the desired velocity $\boldsymbol{v}_{des}$*,*

(2) does not collide with the obstacle, and

(3) is never trapped in a neighborhood of the obstacle.

After designing a control law that solves this problem, we consider an unbounded environment with multiple strictly convex obstacles, in which the following assumption is satisfied.

Assumption 3.1.4. *We define the* ***closest pair of obstacles*** *in the environment as the two obstacles with the shortest distance between their boundaries. We assume that this distance is larger than the diameter* $2r$ *of the robot.*

We confirm that the controller proposed for Problem 3.1.3 guarantees the properties described in the following problem.

Problem 3.1.5. *We consider an unbounded domain that contains a finite number* $m > 1$ *of strictly convex obstacles with arbitrary boundaries described by* $\beta_i(x, y) = 0$*,*

[2] The assumption that the obstacle is *strictly* convex excludes the possibility that its boundary contains straight segments.

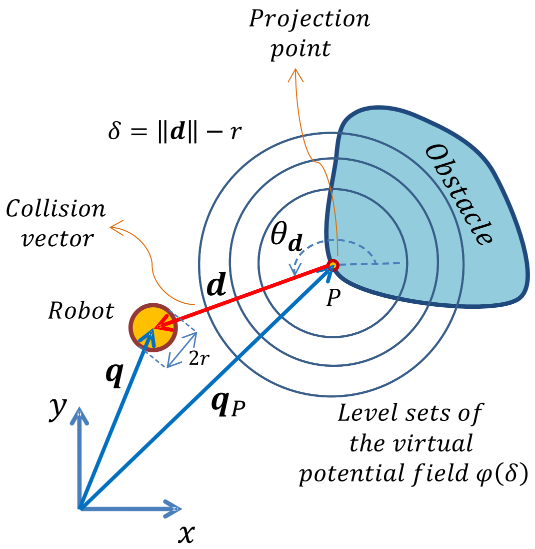

Projection point
$\delta = \|\boldsymbol{d}\| - r$
Obstacle
Collision vector
$\theta_{\boldsymbol{d}}$
$\boldsymbol{d}$
Robot
P
$2r$
$\boldsymbol{q}$
y
$\boldsymbol{q}_P$
Level sets of the virtual potential field $\varphi(\delta)$
x

Figure 3.1: A Schematic Representation of the Robot, an Obstacle, the Projection Point, the Collision Vector, a Virtual Potential Field Constructed by the Robot, and the Associated Global Reference Frame.

where each $\beta_i : \mathbb{R}^2 \mapsto \mathbb{R}$, $i \in \{1, ..., m\}$ is at least twice continuously differentiable. We assume that Assumption 3.1.4 about the distance between obstacles in the domain is satisfied. During its motion, the robot implements the control law designed to solve Problem 3.1.3 for the obstacle that is closest to its current position. We confirm that this control law ensures that the robot:

(1) asymptotically converges to the desired velocity $\boldsymbol{v}_{des}$,

(2) does not collide with any obstacle, and

(3) is never trapped by any set of obstacles.

3.2 Controller Design

The proposed control law is a combination of a regulatory term, which stabilizes the robot's velocity to $\boldsymbol{v}_{des}$, and a repulsive term that is based on the gradient of a virtual potential field.

3.2.1 Definition of the Virtual Potential Field

The robot constructs a virtual potential field φ around the point P. This field is designed to satisfy four properties:

(i) φ is only a function of $\delta := (\|\boldsymbol{d}\| - r) \in \mathbb{R}_{>0}$, the distance between the robot's boundary and the point $\boldsymbol{q}_P$.

(ii) $\varphi(\delta) \to \infty$ as $\delta \to 0$.

(iii) $\frac{d}{d\delta}\varphi(\delta) \to \infty$ as $\delta \to 0$.

(iv) $\varphi(\delta)$ and $\frac{d}{d\delta}\varphi(\delta)$ decrease monotonically to 0 as $\delta \to \delta_c$, and equal zero when $\delta \geq \delta_c$.

Note that by property (i), the potential field has circular level sets around $\boldsymbol{q}_P$, as shown in Fig. 3.1.

To this end, we define the potential field as follows:

$$\varphi(\delta) = \begin{cases} p\frac{\delta}{\delta_c} + \left(\frac{\delta_c}{\delta}\right)^p - (p+1), & 0 < \delta \leq \delta_c, \\ 0, & \delta_c < \delta, \end{cases} \tag{3.1}$$

where p is a strictly positive real constant. We can easily confirm that the function in Eq. 3.1 has properties (i)-(iii). The function also satisfies property (iv), and is therefore continuous and differentiable for every $\delta \in \mathbb{R}_{>0}$. This potential field may introduce extremely large forces into the control law that exceed the saturation limits of the actuators when the robot moves very close to the obstacle's boundary. However, as described in Section 3.3.3, we can enforce an upper bound on the actuation forces if we impose a limit on the robot's speed.

3.2.2 Robot Control Law

The proposed control law is the following combination of a stabilizing term and a repulsive term:

$$\boldsymbol{u} = -\boldsymbol{K}(\dot{\boldsymbol{q}} - \boldsymbol{v}_{des}) - \boldsymbol{K}_R \nabla_{\boldsymbol{d}} \varphi(\delta), \tag{3.2}$$

in which $\boldsymbol{K} = k\boldsymbol{I}$ and $\boldsymbol{K}_R = k_R \boldsymbol{I}$, where k, k_R are positive gains and $\boldsymbol{I} \in \mathbb{R}^{2\times 2}$ is the identity matrix, and $\nabla_{\boldsymbol{d}}\varphi(\delta)$ is the gradient of the potential field with respect to $\boldsymbol{d}$.

Remark 3.2.1. *The gradient of φ with respect to $\boldsymbol{d}$ can be written as:*

$$\nabla_{\boldsymbol{d}}\varphi(\delta) = \frac{d\varphi}{d\delta}\boldsymbol{e}_{\boldsymbol{d}} = \begin{cases} \frac{p}{\delta_c}\left(1 - \left(\frac{\delta_c}{\delta}\right)^p\right)\boldsymbol{e}_{\boldsymbol{d}}, & 0 < \delta \leq \delta_c \\ \boldsymbol{0}, & \delta_c < \delta \end{cases} \tag{3.3}$$

where $\boldsymbol{e}_{\boldsymbol{d}}$ is the unit vector along $\boldsymbol{d}$. The calculation of $\nabla_{\boldsymbol{d}}\varphi$ is provided in Appendix B. Since p and δ_c are known parameters, and we assume that the robot can measure δ and the direction of $\boldsymbol{d}$ (see Section 3.1), the robot can therefore calculate $\nabla_{\boldsymbol{d}}\varphi(\delta)$ using only these local measurements.

Remark 3.2.2. *We emphasize that the control law in Eq. 3.2 relies solely on local measurements: the robot's velocity $\dot{\boldsymbol{q}}$ and the magnitude and direction of the collision vector $\boldsymbol{d}$, all of which can be measured by sensors on-board the robot. The robot does not need global position information or knowledge about the locations and geometric properties of the obstacles.*

3.3 Analysis of Robot Dynamics for Single-Obstacle Case

We now investigate the robot's closed-loop dynamics with the control law in Eq. 3.2 and prove that this control law achieves the three objectives described in Problem 3.1.3. First, we define four terms that will be used in our analysis.

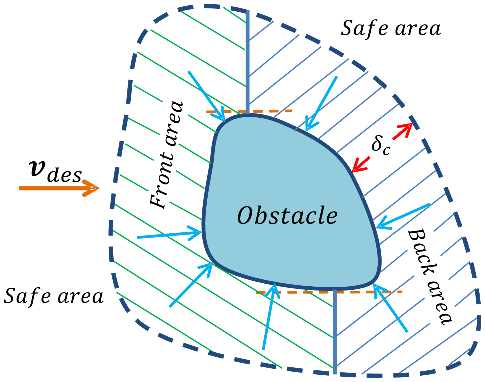


Figure 3.2: Illustration of an Obstacle's *Front* and *Back* Areas as well as the *Safe* Area. The Dashed Orange Lines Are Parallel to the Direction of the Desired Velocity, and the Solid Blue Lines Are Normal to the Dashed Orange Lines. The Blue Arrows Illustrate the Gradient of the Potential Field.

Definition 3.3.1. *The **free space** is the subset of the domain that excludes the obstacle's boundary and interior.*

Definition 3.3.2. *The **obstacle's front area** is the subset of the free space in which* $\delta \in (0, \delta_c]$ *and* $\boldsymbol{v}_{des}^T \nabla_{\boldsymbol{d}} \varphi \geq 0$.

Definition 3.3.3. *The **obstacle's back area** is the subset of the free space in which* $\delta \in (0, \delta_c]$ *and* $\boldsymbol{v}_{des}^T \nabla_{\boldsymbol{d}} \varphi < 0$.

Definition 3.3.4. *The **safe area** is the subset of the free space that excludes the obstacle's front and back areas.*

The areas defined above are illustrated for an arbitrary strictly convex obstacle in Fig. 3.2.

3.3.1 Velocity Convergence Analysis

We can write the closed-loop dynamics of the robot with the proposed control law as:

$$\ddot{\boldsymbol{q}} + \boldsymbol{K}(\dot{\boldsymbol{q}} - \boldsymbol{v}_{des}) + \boldsymbol{K}_R \nabla_{\boldsymbol{d}} \varphi(\delta) = \boldsymbol{0}. \tag{3.4}$$

By setting $\ddot{\boldsymbol{q}} = \dot{\boldsymbol{q}} = \boldsymbol{0}$ in Eq. 3.4, we obtain $\boldsymbol{K}\boldsymbol{v}_{des} = \boldsymbol{K}_R \boldsymbol{\nabla}_{\boldsymbol{d}}\varphi(\delta)$. Since the obstacle is *strictly* convex, given any two distinct points at the same distance δ from the obstacle, the direction of the gradient $\boldsymbol{\nabla}_{\boldsymbol{d}}\varphi(\delta)$ (i.e., the direction of $\boldsymbol{e}_{\boldsymbol{d}}$) at these points cannot be identical. Also, we can confirm that $\frac{d\varphi}{d\delta}$ in Eq. 3.3 is a strictly monotonic function for $\delta \in (0, \delta_c)$. Thus, the equation $\boldsymbol{K}\boldsymbol{v}_{des} = \boldsymbol{K}_R \boldsymbol{\nabla}_{\boldsymbol{d}}\varphi(\delta)$ has unique solutions for δ and $\boldsymbol{e}_{\boldsymbol{d}}$, which we denote by δ_e and $\boldsymbol{e}_{\boldsymbol{d}_e}$, respectively. Consequently, Eq. 3.4 has a unique equilibrium point at which

$$\delta_e = \delta_c \left(1 + \frac{k \left\|\boldsymbol{v}_{des}\right\|}{pk_R}\right)^{-1/(p+1)}, \qquad \boldsymbol{e}_{\boldsymbol{d}_e} = -\boldsymbol{e}_{\boldsymbol{v}_{des}}, \tag{3.5}$$

where $\boldsymbol{e}_{\boldsymbol{v}_{des}}$ is the unit vector along $\boldsymbol{v}_{des}$. We can check that $\delta_e \in (0, \delta_c)$. The repulsive vector field $-\nabla_{\boldsymbol{d}}\varphi$ has a component in the opposite direction of $\boldsymbol{v}_{des}$ everywhere in the front area of the obstacle, and it has a component in the same direction as $\boldsymbol{v}_{des}$ everywhere in the back area. Thus, the position where the term $\boldsymbol{K}_R\nabla_{\boldsymbol{d}}\varphi$ negates the term $-\boldsymbol{K}\boldsymbol{v}_{des}$ in Eq. 3.4 must be in the front area, and so the equilibrium where the robot stops at a distance δ_e from the obstacle must be in this area (see Fig. 3.3).

Eq. 3.4 also has an invariant set $\mathcal{E}$ that is defined as

$$\mathcal{E} = \left\{\boldsymbol{q} \in \mathbb{R}^2 \ , \ \dot{\boldsymbol{q}} \in \mathbb{R}^2 \mid \dot{\boldsymbol{q}} = \boldsymbol{v}_{des} \ , \ \nabla_{\boldsymbol{d}}\varphi(\delta) = \boldsymbol{0}\right\}. \tag{3.6}$$

From Eq. 3.3, $\nabla_{\boldsymbol{d}}\varphi = \boldsymbol{0}$ implies that $\delta \geq \delta_c$. This invariant set has no intersection with the obstacle's front area, since asymptotic convergence to the desired velocity $\boldsymbol{v}_{des}$ and a monotonic decrease in the value of the potential field $\varphi(\delta)$ as $\delta \to \delta_c$ (property (iv) of φ) cannot occur simultaneously in the front area.

The stability characteristics of the unique equilibrium point in Eq. 3.5 and the invariant set in Eq. 3.6 are discussed in the next two theorems.

Theorem 3.3.5. *Consider the unique equilibrium point of Eq. 3.4 for which the robot is stationary at distance δ_e, given in Eq. 3.5, from the boundary of the obstacle in the*

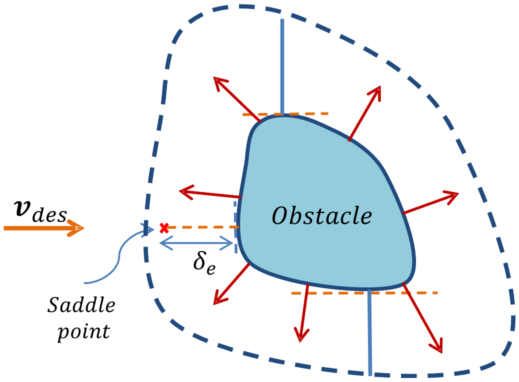


Figure 3.3: Illustration of Repulsive Vector Field $-\nabla_{\boldsymbol{d}}\varphi$ (Red Arrows) and the Unique Equilibrium Point at Distance δ_e from the Obstacle, Given by Eq. 3.5.

obstacle's front area. This equilibrium is a saddle point.

Proof. We use Lyapunov's indirect method to investigate the stability properties of this equilibrium. Toward this end, we define the state vector as $\boldsymbol{X} = (\boldsymbol{q}^T, \dot{\boldsymbol{q}}^T)^T \in \mathbb{R}^4$ and linearize Eq. 3.4 about the equilibrium point, obtaining the following equation:

$$\dot{\boldsymbol{X}} = \begin{bmatrix} \boldsymbol{0}_{2\times 2} & \boldsymbol{I}_{2\times 2} \\ -\boldsymbol{K}_R \frac{\partial}{\partial \boldsymbol{q}} \nabla_{\boldsymbol{d}}\varphi(\delta) & -\boldsymbol{K} \end{bmatrix} \boldsymbol{X}, \tag{3.7}$$

where $\delta \equiv \delta_e$. By *Lemma 3.5* in (Koditschek, 1989), we know that the linear system in Eq. 3.7 has the stability properties of the system described by $\dot{\boldsymbol{q}} = -\frac{\partial}{\partial \boldsymbol{q}} \nabla_{\boldsymbol{d}}\varphi(\delta_e)\boldsymbol{q}$. In addition, since $\boldsymbol{q} = \boldsymbol{d} + \boldsymbol{q}_P$ and $\varphi = \varphi(\delta)$, where $\delta = (\|\boldsymbol{d}\| - r)$, we can show that $\nabla_{\boldsymbol{d}}\varphi = \nabla_{\boldsymbol{q}}\varphi$ (this equation is similar to Equation (7) in (Tanner *et al.*, 2003) and is proved in Appendix B). Therefore, we have that $\dot{\boldsymbol{q}} = -\frac{\partial}{\partial \boldsymbol{q}} \nabla_{\boldsymbol{d}}\varphi(\delta_e)\boldsymbol{q} = -\nabla^2\varphi(\delta_e)\boldsymbol{q}$, where $\nabla^2\varphi(\delta_e)$ is the Hessian of φ at the equilibrium point. The stability properties of Eq. 3.7 are thus determined by the eigenvalues of $\nabla^2\varphi(\delta_e)$, which we characterize in the following lemma.

Lemma 3.3.6. *The determinant of $\nabla^2\varphi(\delta)$ is strictly negative for all points $\boldsymbol{q} \in \mathbb{R}^2$ such that $\delta \in (0, \delta_c)$, and consequently, the eigenvalues of $\nabla^2\varphi(\delta)$ have opposite signs.*

Proof. The Hessian of φ can be calculated as

$$\nabla^2\varphi(\delta) \;=\; \frac{\partial}{\partial \boldsymbol{q}}\left(\nabla_{\boldsymbol{d}}\varphi(\delta)\right) \;=\; \frac{\partial}{\partial \boldsymbol{q}}\left(\varphi'(\delta)\boldsymbol{e}_{\boldsymbol{d}}\right), \tag{3.8}$$

where $\varphi'(\delta) = \frac{d\varphi}{d\delta}$. Applying the fact that $\nabla_{\boldsymbol{d}}\varphi = \nabla_{\boldsymbol{q}}\varphi$, Eq. 3.8 can be written as

$$\nabla^2\varphi(\delta) \;=\; \varphi''(\delta)\boldsymbol{e}_{\boldsymbol{d}}\boldsymbol{e}_{\boldsymbol{d}}^T + \varphi'(\delta)\left(\frac{\partial \boldsymbol{e}_{\boldsymbol{d}}}{\partial \boldsymbol{q}}\right), \tag{3.9}$$

where $\varphi''(\delta) = \frac{d^2\varphi}{d\delta^2}$. By the chain rule, the partial derivative in Eq. 3.9 can be expressed as

$$\frac{\partial \boldsymbol{e}_{\boldsymbol{d}}}{\partial \boldsymbol{q}} = \frac{\partial \boldsymbol{e}_{\boldsymbol{d}}}{\partial \boldsymbol{d}}\frac{\partial \boldsymbol{d}}{\partial \boldsymbol{q}}. \tag{3.10}$$

Since $\boldsymbol{d} = \boldsymbol{q} - \boldsymbol{q}_P$, we have that $\frac{\partial \boldsymbol{d}}{\partial \boldsymbol{q}} = \boldsymbol{I}$. Also, given that $\boldsymbol{e}_{\boldsymbol{d}} = [\cos(\theta_{\boldsymbol{d}})\;\; \sin(\theta_{\boldsymbol{d}})]^T$, we can confirm that

$$\frac{\partial \boldsymbol{e}_{\boldsymbol{d}}}{\partial \boldsymbol{d}} = \frac{1}{\delta}\begin{bmatrix} \sin(\theta_{\boldsymbol{d}})^2 & -\cos(\theta_{\boldsymbol{d}})\sin(\theta_{\boldsymbol{d}}) \\ -\cos(\theta_{\boldsymbol{d}})\sin(\theta_{\boldsymbol{d}}) & \cos(\theta_{\boldsymbol{d}})^2 \end{bmatrix}. \tag{3.11}$$

Using Eqs. 3.10 and 3.11, Eq. 3.9 can be rewritten as

$$\begin{aligned} \nabla^2\varphi(\delta) \;=\; & \varphi''(\delta)\begin{bmatrix} \cos(\theta_{\boldsymbol{d}})^2 & \cos(\theta_{\boldsymbol{d}})\sin(\theta_{\boldsymbol{d}}) \\ \cos(\theta_{\boldsymbol{d}})\sin(\theta_{\boldsymbol{d}}) & \sin(\theta_{\boldsymbol{d}})^2 \end{bmatrix} \\ & + \frac{\varphi'(\delta)}{\delta}\begin{bmatrix} \sin(\theta_{\boldsymbol{d}})^2 & -\cos(\theta_{\boldsymbol{d}})\sin(\theta_{\boldsymbol{d}}) \\ -\cos(\theta_{\boldsymbol{d}})\sin(\theta_{\boldsymbol{d}}) & \cos(\theta_{\boldsymbol{d}})^2 \end{bmatrix}. \end{aligned} \tag{3.12}$$

Then, we can express the determinant of the Hessian as

$$\det(\nabla^2\varphi(\delta)) \;=\; \frac{1}{\delta}\varphi'(\delta)\varphi''(\delta). \tag{3.13}$$

We can determine from Eq. 3.1 that $\varphi'(\delta)$ and $\varphi''(\delta)$ are strictly negative and strictly positive, respectively, for $\delta \in (0, \delta_c)$. Hence, $\det(\nabla^2\varphi(\delta))$ is strictly negative for any point $\boldsymbol{q}$ that is at a distance $\delta \in (0, \delta_c)$ from a strictly convex obstacle. □

Since $\det(\nabla^2\varphi)$ is strictly negative, the eigenvalues of $\nabla^2\varphi$ are both non-zero and have opposite signs, and consequently the equilibrium of the system described by $\dot{\boldsymbol{q}} = -\nabla^2\varphi(\delta_e)\boldsymbol{q}$ is a saddle point. As explained in the text preceding Lemma 3.3.6, this implies that the equilibrium of the system in Eq. 3.7 is also a saddle point. Therefore, the equilibrium of Eq. 3.4 for which the robot is stationary at distance δ_e from the obstacle is a saddle point. We can thus conclude that the robot can only reach this equilibrium if its initial position is in a set of measure zero. In practice, the robot will not be initialized precisely in this set, and so it will never stop at the location of the saddle point. □

Theorem 3.3.7. *The invariant set $\mathcal{E}$ described in Eq. 3.6 is locally asymptotically stable, and the obstacle's back area is a subset of its basin of attraction.*

Proof. We cannot use Lyapunov's indirect method to study the stability of the invariant set $\mathcal{E}$ due to the following argument. In the set $\mathcal{E}$, we have that $\delta \geq \delta_c$, and consequently $\varphi(\delta) = 0$ and $d\varphi(\delta)/d\delta = 0$ in this set. The linearization of Eq. 3.4 about each point in the set $\mathcal{E}$ is given by Eq. 3.7 with $\delta \geq \delta_c$. As a result, the first two columns of the matrix in Eq. 3.7 are both columns of zeros, and therefore the matrix has two zero eigenvalues. Thus, we cannot determine the stability characteristics of the closed-loop system from its linearization (Khalil, 1996).

Instead, we use LaSalle's invariance principle for this case. Toward this end, we define the velocity error $\boldsymbol{s} = \dot{\boldsymbol{q}} - \boldsymbol{v}_{des}$. Since the desired velocity is constant, we have that $\ddot{\boldsymbol{q}} = \dot{\boldsymbol{s}}$. Then, the closed-loop dynamics in Eq. 3.4 can be rewritten as:

$$\dot{\boldsymbol{s}} + \boldsymbol{K}\boldsymbol{s} + \boldsymbol{K}_R\nabla_{\boldsymbol{d}}\varphi = \boldsymbol{0}. \tag{3.14}$$

We consider the following Lyapunov function:

$$V = \frac{1}{2}\boldsymbol{s}^T\boldsymbol{s} + k_R\varphi(\delta). \tag{3.15}$$

This function is positive over the entire state space and equals zero at each point in the set $\mathcal{E}$, since $\boldsymbol{s} = \boldsymbol{0}$ and $\varphi(\delta) = 0$ in this set. The time derivative of this function is:

$$\dot{V} = -\boldsymbol{s}^T \boldsymbol{K} \boldsymbol{s} - \boldsymbol{s}^T \boldsymbol{K}_R \nabla_{\boldsymbol{d}} \varphi + k_R (\nabla_{\boldsymbol{d}} \varphi)^T \dot{\boldsymbol{d}}, \tag{3.16}$$

where we have written $\dot{\varphi} = (\nabla_{\boldsymbol{d}} \varphi)^T \dot{\boldsymbol{d}}$ in the last term using the chain rule. To simplify Eq. 3.16, we write $\dot{\boldsymbol{d}}$ as $\dot{\boldsymbol{d}} = \dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_P$, in which $\dot{\boldsymbol{q}}_P$ is the time derivative of the position of the projection point (see Fig. 3.1). Since the projection point is constrained to move along the boundary of the obstacle, its velocity $\dot{\boldsymbol{q}}_P$ is always tangent to this boundary. Moreover, the gradient of the potential field φ is normal to the boundary. Hence, we can conclude that $(\nabla_{\boldsymbol{d}} \varphi)^T \dot{\boldsymbol{q}}_P = 0$. Thus, using the relation $\boldsymbol{s} = \dot{\boldsymbol{q}} - \boldsymbol{v}_{des}$, Eq. 3.16 is simplified to:

$$\dot{V} = -\boldsymbol{s}^T \boldsymbol{K} \boldsymbol{s} + k_R \boldsymbol{v}_{des}^T \nabla_{\boldsymbol{d}} \varphi. \tag{3.17}$$

As stated in Definition 3.3.3, the second term on the right-hand side of this equation is negative in the obstacle's back area, and thus $\dot{V}$ is negative definite over the entire back area of the obstacle. This means that the invariant set $\mathcal{E}$ is locally asymptotically stable, and a set defined as $\Omega := \{\boldsymbol{X} \in \mathbb{R}^4 \mid V \leq c,\ c > 0\}$, which contains the obstacle's back area, is the simplest estimate of the basin of attraction for $\mathcal{E}$. The set Ω consists of all trajectories with a bounded initial velocity that start in or enter the obstacle's back area. □

3.3.2 Collision Avoidance Analysis

We now prove that the robot will never collide with the obstacle in either its front area or back area. For the case where the robot is in the back area, the following corollary from Theorem 3.3.7 ensures collision avoidance:

Corollary 3.3.8. *Since $\dot{V}$ is negative everywhere in the obstacle's back area, V can never become unbounded in this area. This implies that φ never blows up to infinity in the back area. Hence, δ never approaches zero in this region, meaning that the robot never collides with the obstacle when it is in the back area.*

Next, we analyze the case where the robot is in the obstacle's front area. For this purpose, we study the dynamics of the robot in a different coordinate system, illustrated in Fig. 3.4. Note that the vectors denoted by $\boldsymbol{e}$ in the figure are unit vectors. First, we decompose the robot's velocity $\dot{\boldsymbol{q}}$ into the sum of $\dot{\boldsymbol{q}}_P$, the velocity of the projection point on the obstacle's boundary, and $\dot{\boldsymbol{d}}$, the robot's velocity relative to the projection point. We describe $\dot{\boldsymbol{q}}_P$ in a tangential-normal coordinate system (Meriam and Kraige, 2012), in which $\xi \in \mathbb{R}$ denotes the scalar displacement of the projection point along the obstacle's boundary, and $\rho \in \mathbb{R}_{>0}$ denotes the radius of curvature of the boundary. In addition, we describe $\dot{\boldsymbol{d}}$ in a polar coordinate system (Meriam and Kraige, 2012), in which (as defined previously) $\delta = (\|\boldsymbol{d}\| - r) \in \mathbb{R}_{\geq 0}$ is the distance between the robot's boundary and the projection point, and $\theta_{\boldsymbol{d}} \in [-\pi, \pi]$ rad is the angle of the vector $\boldsymbol{d}$ in the global reference frame. Using the facts that $\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}_P + \dot{\boldsymbol{d}}$, $\boldsymbol{e}_\delta = \boldsymbol{e}_{\boldsymbol{d}}$, and $\dot{\boldsymbol{q}}_P$ is always tangent to the obstacle's boundary, the robot's velocity can be written in the new coordinate system as:

$$\dot{\boldsymbol{q}} = (\dot{\delta})\boldsymbol{e}_\delta + \Big((\delta + r)\dot{\theta}_{\boldsymbol{d}}\Big)\,\boldsymbol{e}_{\theta_{\boldsymbol{d}}} + (\dot{\xi})\boldsymbol{e}_t. \tag{3.18}$$

Therefore, the robot's acceleration is:

$$\ddot{\boldsymbol{q}} = \Big(\ddot{\delta} - (\delta + r)\dot{\theta}_{\boldsymbol{d}}^2\Big)\,\boldsymbol{e}_\delta + \Big((\delta + r)\ddot{\theta}_{\boldsymbol{d}} + 2\dot{\delta}\dot{\theta}_{\boldsymbol{d}}\Big)\,\boldsymbol{e}_{\theta_{\boldsymbol{d}}} + (\ddot{\xi})\boldsymbol{e}_t + \left(\frac{\dot{\xi}^2}{\rho}\right)\boldsymbol{e}_n. \tag{3.19}$$

Moreover, since the collision vector $\boldsymbol{d}$ always points in the direction of the normal to the boundary, we can conclude that $\boldsymbol{e}_t = \boldsymbol{e}_{\theta_{\boldsymbol{d}}}$ and $\boldsymbol{e}_n = -\boldsymbol{e}_\delta$. Substituting the expressions for $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ defined in Eq. 3.18 and Eq. 3.19 into Eq. 3.4, we can write

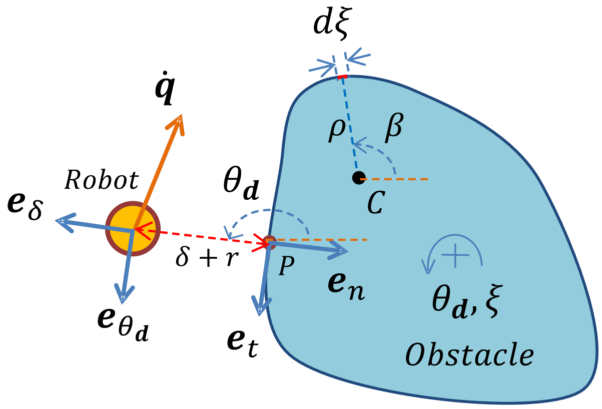


Figure 3.4: Illustration of the Coordinate Systems Used to Derive Eqs. 3.20 and 3.21.

the resulting equations of motion along the $\boldsymbol{e}_\delta$ and $\boldsymbol{e}_{\theta_{\boldsymbol{d}}}$ directions as Eq. 3.20 and Eq. 3.21, respectively:

$$\ddot{\delta} - (\delta + r)\dot{\theta}_{\boldsymbol{d}}^2 - \frac{\dot{\xi}^2}{\rho} + k\dot{\delta} + k_R\varphi'(\delta) - kv_{des}\cos(\theta_{\boldsymbol{d}}) = 0, \tag{3.20}$$

$$(\delta + r)\ddot{\theta}_{\boldsymbol{d}} + 2\dot{\delta}\dot{\theta}_{\boldsymbol{d}} + \ddot{\xi} + k(\delta + r)\dot{\theta}_{\boldsymbol{d}} + k\dot{\xi} + kv_{des}\sin(\theta_{\boldsymbol{d}}) = 0, \tag{3.21}$$

in which $v_{des} = \|\boldsymbol{v}_{des}\|$. Note that the repulsive force $-\varphi'(\delta)\boldsymbol{e}_{\boldsymbol{d}}$ shows up only in Eq. 3.20, since $\boldsymbol{e}_{\boldsymbol{d}} = \boldsymbol{e}_\delta$.

Remark 3.3.9. *The robot is in the obstacle's back area when* $\cos(\theta_{\boldsymbol{d}}) > 0$*, and it is in the front area when* $\cos(\theta_{\boldsymbol{d}}) \leq 0$.

The next theorem uses Eq. 3.20 and Eq. 3.21 to prove that the robot will never collide with the obstacle in its front area.

Theorem 3.3.10. *If the robot's trajectory starts anywhere in the obstacle's front area, then the robot will never collide with the obstacle's boundary in this region.*

Proof. We consider the following function:

$$\begin{aligned} W =& \frac{1}{2}\left(\dot{\delta}^2 + \left((\delta + r)\dot{\theta}_{\boldsymbol{d}} + \dot{\xi}\right)^2\right) + k_R\varphi(\delta) \\ & - kv_{des}\left((\delta + r)\cos(\theta_{\boldsymbol{d}}) - \int_0^{\theta_{\boldsymbol{d}}} \rho\sin(\sigma)d\sigma\right). \end{aligned} \tag{3.22}$$

To confirm that this function is positive over all $\boldsymbol{\theta_d}$ in the obstacle's front area, i.e. $\boldsymbol{\theta_d} \in [-\pi, -\pi/2] \cup [\pi/2, \pi]$ rad, we only have to prove that $(\delta \cos(\boldsymbol{\theta_d}) - \int_0^{\boldsymbol{\theta_d}} \rho \sin(\sigma) d\sigma) \leq 0$ for all $\boldsymbol{\theta_d}$ in this set. From Remark 3.3.9, we see that the term $\delta \cos(\boldsymbol{\theta_d}) \leq 0$ for all $\boldsymbol{\theta_d}$ in the front area. In addition, by Definition 6.2 in (Khalil, 1996), the integral $\int_0^{\boldsymbol{\theta_d}} \rho \sin(\sigma) d\sigma \geq 0$ for any $\boldsymbol{\theta_d} \in [-\pi, \pi]$ rad, since ρ is always positive and $\sin(\boldsymbol{\theta_d})$ belongs to the sector $[0, \pi/4]$ for $\boldsymbol{\theta_d} \in [-\pi, \pi]$ rad (and therefore for $\boldsymbol{\theta_d} \in [-\pi, -\pi/2] \cup [\pi/2, \pi]$ rad). Therefore, $(\delta \cos(\boldsymbol{\theta_d}) - \int_0^{\boldsymbol{\theta_d}} \rho \sin(\sigma) d\sigma) \leq 0$ for all $\boldsymbol{\theta_d}$ in the front area, and hence W is positive over such $\boldsymbol{\theta_d}$. We note that we cannot derive a closed-form solution for the integral $\int_0^{\boldsymbol{\theta_d}} \rho \sin(\sigma) d\sigma$, since ρ changes with $\boldsymbol{\theta_d}$ for arbitrary strictly convex obstacles, and the obstacle shape in our scenario is unknown.

The time derivative of the function W along the trajectories of the system in Eq. 3.20 and Eq. 3.21 is given by:

$$\begin{aligned} \dot{W} = & -k \left(\dot{\delta}^2 + (\delta + r)^2 \dot{\boldsymbol{\theta}}_{\boldsymbol{d}}^2 \right) + \dot{\delta} \frac{\dot{\xi}^2}{\rho} - \dot{\xi} \dot{\delta} \dot{\boldsymbol{\theta}}_{\boldsymbol{d}} \\ & - k v_{des} \left(\dot{\xi} \sin(\boldsymbol{\theta_d}) - \frac{d}{dt} \int_0^{\boldsymbol{\theta_d}} \rho \sin(\sigma) d\sigma \right). \end{aligned} \tag{3.23}$$

We define β as the angle of the direction of the normal to the boundary in the global reference frame. For an infinitesimal change in the projection point displacement ξ, we have that $d\xi = \rho d\beta$, where the radius of curvature ρ is approximated as constant. This implies that $\dot{\xi} = \rho \dot{\beta}$. Moreover, since $\boldsymbol{e}_\delta$ is always normal to the boundary, we can conclude that $d\boldsymbol{\theta_d} = d\beta$, and consequently, $\dot{\beta} = \dot{\boldsymbol{\theta}}_{\boldsymbol{d}}$. Using the relation $\dot{\xi} = \rho \dot{\boldsymbol{\theta}}_{\boldsymbol{d}}$, we can reduce Eq. 3.23 to the following expression:

$$\dot{W} = -k \left(\dot{\delta}^2 + (\delta + r)^2 \dot{\boldsymbol{\theta}}_{\boldsymbol{d}}^2 \right) - k v_{des} \left(\rho \dot{\boldsymbol{\theta}}_{\boldsymbol{d}} \sin(\boldsymbol{\theta_d}) - \frac{d}{dt} \int_0^{\boldsymbol{\theta_d}} \rho \sin(\sigma) d\sigma \right). \tag{3.24}$$

We now define $g(\boldsymbol{\theta_d}) := \int_0^{\boldsymbol{\theta_d}} \rho \sin(\sigma) d\sigma$. By the chain rule, the time derivative of $g(\boldsymbol{\theta_d})$ can be written as $\frac{d}{dt} g(\boldsymbol{\theta_d}) = \frac{d}{d\boldsymbol{\theta_d}} g(\boldsymbol{\theta_d}) \dot{\boldsymbol{\theta}}_{\boldsymbol{d}}$, where $\frac{d}{d\boldsymbol{\theta_d}} g(\boldsymbol{\theta_d}) = \rho \sin(\boldsymbol{\theta_d})$. This leads to the cancellation of the two terms in the second set of parentheses in Eq. 3.24, and $\dot{W}$ is simplified to:

$$\dot{W} = -k\left(\dot{\delta}^2 + (\delta + r)^2 \dot{\theta}_{\boldsymbol{d}}^2\right), \tag{3.25}$$

from which we can conclude that $\dot{W} \leq 0$. Therefore, W never becomes unbounded in the front area, which implies that φ remains bounded in this region. By property (ii) of φ, this shows that the distance δ never approaches zero in the front area, and hence the robot never collides with the obstacle in this region. □

The next theorem, which addresses the evolution of robot trajectories that begin in the obstacle's front area, completes our analysis of collision avoidance.

Theorem 3.3.11. *Almost all robot trajectories that start in the obstacle's front area will eventually leave this region and enter the back area or the safe area.*

Proof. From Theorem 3.3.5, the only equilibrium point in the front area is a saddle, which does not attract any trajectories in this area except for trajectories that start in a particular set of measure zero. Furthermore, since there are no other equilibria in the front area, we can apply the *Index Lemma* in (Khalil, 1996) to Eq. 3.14 and conclude that there is no limit cycle in this area as well. Therefore, the unstable trajectories in the front area, which emanate from the saddle point, must cross into the back area. By Theorem 3.3.7, there exists an asymptotically stable invariant set in the back area that attracts these trajectories. □

3.3.3 A Bound on The Repulsive Term in the Control Input

As stated in property (iii) in Section 3.2.1, the derivative of the potential field φ goes to infinity when the robot's distance δ from the obstacle approaches zero. Thus, when the robot is very close to the obstacle (i.e., δ is small), the repulsive term in the control input Eq. 3.2 could become too large to implement in practice. In this section, we establish an upper bound on this term by incorporating realistic constraints on the robot's initial velocity and sensing range.

We consider the line that is parallel to the direction of the desired velocity and passes through the saddle equilibrium point, as shown in Fig. 3.3. If the robot's initial position is located on this line, and its initial velocity is parallel to this line, then both the velocity stabilizing force $-\boldsymbol{K}(\dot{\boldsymbol{q}} - \boldsymbol{v}_{des})$ and the repulsive force will be along this line at the beginning of its motion and for all future time, since there will be no other vector fields to drive the robot off this direction. This leads to one-dimensional motion of the robot along this line. Moreover, for a given initial robot speed, the velocity stabilizing force has the largest component that directly opposes the repulsive force when the robot is on this line, compared to when it is anywhere else in the obstacle's front area. Thus, the minimum feasible value for δ in the front area is achieved on this line.

When the robot moves only along this line in the front area, we have that $\ddot{\theta}_{\boldsymbol{d}}, \dot{\theta}_{\boldsymbol{d}}, \ddot{\xi}, \dot{\xi} = 0$ and $\theta_{\boldsymbol{d}} = \pi$. Substituting these values into Eqs. 3.20 and 3.21, the equation of the robot's one-dimensional motion along this line in the front area is given by:

$$\ddot{\delta} + k\dot{\delta} + k_R\varphi'(\delta) + kv_{des} = 0. \tag{3.26}$$

Theorem 3.3.13 below proves the existence of a lower bound on δ by comparing the time response of Eq. 3.26, which is denoted by $\delta(t)$, to that of the following equation,

$$\ddot{\varrho} + k\dot{\varrho} = h(\varrho), \tag{3.27}$$

where

$$h(\varrho) := -a\varrho + b, \qquad a, b \in \mathbb{R}_{>0}. \tag{3.28}$$

We first state the following Lemma, which describes the conditions that guarantee the existence of a positive lower bound for the time response of Eq. 3.27, denoted by $\varrho(t)$.

Lemma 3.3.12. *Given $\varrho_0 := \varrho(0) \in \mathbb{R}_{>0}$[3] and $w_0 := \dot{\varrho}(0)$ as the initial conditions for Eq. 3.27, and $v_{max} \in \mathbb{R}_{>0}$ as a bound for w_0, i.e. $|w_0| \leq v_{max}$, there exists a strictly positive number γ that satisfies $\varrho(t) \geq \gamma$, $\forall t \in [0, \infty)$, if a and b in Eq. 3.28 are large enough.*

Proof. See Appendix C. □

Theorem 3.3.13. *Given $\delta_0 := \delta(0)$ and $v_0 := \dot{\delta}(0)$ as the robot's initial distance from the obstacle and its initial speed in Eq. 3.26, respectively, and also assuming $|v_0| \leq v_{max}$, there exists a lower bound on the robot's distance from the obstacle. If the robot starts its motion in the safe area, this bound will be uniform and depend on δ_c and v_{max}.*

Proof. We rewrite Eq. 3.26 in the following form:

$$\ddot{\delta} + k\dot{\delta} = -k_R\varphi'(\delta) - kv_{des}. \tag{3.29}$$

The right-hand side of Eq. 3.29 is the orange curve in Fig. 3.5. From this plot, we can see that there is at least one function $h(\delta)$ in the form of Eq. 3.28 (the green straight line in Fig. 3.5) that satisfies the following condition,

$$h(\delta) \leq -k_R\varphi'(\delta) - kv_{des}, \qquad \forall \delta \in \mathbb{R}_{>0}, \tag{3.30}$$

for the type of potential field defined by Eq. 3.1. Hence, we can write the following differential inequality:

$$\ddot{\delta} + k\dot{\delta} \geq -a\delta + b, \qquad \delta(0) = \delta_0, \quad \dot{\delta}(0) = v_0. \tag{3.31}$$

Now, considering the system in Eq. 3.27 with the initial conditions set as $\varrho_0 = \delta_0,\ w_0 = v_0$, and defining

$$\chi_\varrho := \dot{\varrho} + k\varrho, \qquad \chi_\delta := \dot{\delta} + k\delta, \tag{3.32}$$

[3] We assume that ϱ_0 is positive, since we want to compare $\varrho(t)$ with $\delta(t)$, which we proved is always positive in Theorem 3.3.10.

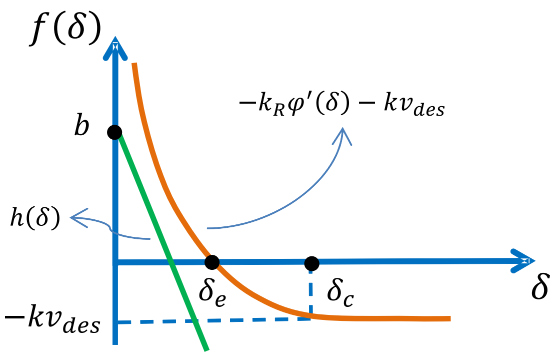


Figure 3.5: Illustration of the Right-Hand Side of Eq. 3.29 Compared to a Function in the Form of Eq. 3.28 with ϱ Replaced by δ.

we rewrite Eq. 3.27 and Eq. 3.31 as

$$\dot{\chi}_\varrho = h(\varrho), \qquad \dot{\chi}_\delta \geq h(\delta). \tag{3.33}$$

Using the *comparison lemma* (Khalil, 1996), we obtain

$$\chi_\delta \geq \chi_\varrho, \qquad \forall t \in [0, \infty). \tag{3.34}$$

Furthermore, using the expressions for χ_δ and χ_ϱ in Eq. 3.32, we can rewrite Eq. 3.34 as

$$\dot{\delta} - \dot{\varrho} \geq -k(\delta - \varrho). \tag{3.35}$$

Using the *comparison lemma* again, we obtain

$$\delta(t) \geq (\delta_0 - \varrho_0)\, e^{-kt} + \varrho(t), \qquad \forall t \in [0, \infty), \tag{3.36}$$

and taking into account the fact that $\delta_0 = \varrho_0$, we conclude that

$$\delta(t) \geq \varrho(t), \qquad \forall t \in [0, \infty). \tag{3.37}$$

Finally, invoking Lemma 3.3.12, we can write

$$\delta(t) \geq \gamma, \tag{3.38}$$

which gives a lower bound for $\delta(t)$ and completes the proof. □

If we assume that the robot starts its motion in the safe area, we can replace δ_0 with δ_c in all the calculations and obtain a closed-form solution for γ based on the procedure in C. Such a bound is a function of δ_c and v_{max}, and this bound is uniform with respect to the robot's initial condition.

To conclude this section, we take into account the fact that $-\varphi'(\delta)$ is a decreasing function with respect to δ, which allows us to establish an upper bound for the repulsive term based on the derived lower bound on $\delta(t)$ as

$$\|-k_R\varphi'(\delta)\boldsymbol{e_d}\| \leq -k_R\varphi'(\gamma). \tag{3.39}$$

3.4 Analysis of Robot Dynamics for Multiple-Obstacle Case

In this section, we design a control law based on Eq. 3.2, which was developed for an environment with a single obstacle, and demonstrate that it achieves the three objectives described in Problem 3.1.5 for an environment that contains multiple strictly convex obstacles. Our solution is to define a switching control law, in which the robot applies the control law Eq. 3.2 for the closest obstacle that it detects in its sensing range at each time instant. This control law is a discontinuous function because the repulsive term in the control input Eq. 3.2 undergoes a sudden change in its direction whenever the robot crosses the *switching surface* between two obstacles, which is the loci of all points that are equidistant from the obstacles, as illustrated in Fig. 3.6. If there are m disjoint obstacles in the robot's sensing range, the control law is written as:

$$\begin{aligned}
&\boldsymbol{u} = -\boldsymbol{K}(\dot{\boldsymbol{q}} - \boldsymbol{v}_{des}) - \boldsymbol{K}_R\nabla_{\boldsymbol{d^*}}\varphi(\delta^*),\\
&\delta^* = \min_{i\in\{1,\ldots,m\}}\{\delta_i\},
\end{aligned} \tag{3.40}$$

where $\boldsymbol{d}^*$ is the collision vector associated with the closest obstacle. The closed-loop dynamics of the robot with control law Eq. 3.40 can be written as:

$$\ddot{\boldsymbol{q}} + \boldsymbol{K}(\dot{\boldsymbol{q}} - \boldsymbol{v}_{des}) + \boldsymbol{K}_R \nabla_{\boldsymbol{d}^*} \varphi(\delta^*) = \boldsymbol{0}. \tag{3.41}$$

Defining the state vector $\boldsymbol{X} = (\boldsymbol{X}_1^T, \boldsymbol{X}_2^T)^T \in \mathbb{R}^4$, where $\boldsymbol{X}_1 = \boldsymbol{q}$ and $\boldsymbol{X}_2 = \dot{\boldsymbol{q}}$, we can rewrite Eq. 3.41 in state-space form as

$$\dot{\boldsymbol{X}} = \boldsymbol{f}^*(\boldsymbol{X}) := \begin{bmatrix} \boldsymbol{X}_2 \\ -\boldsymbol{K}(\boldsymbol{X}_2 - \boldsymbol{v}_{des}) - \boldsymbol{K}_R \nabla_{\boldsymbol{d}^*} \varphi(\delta^*) \end{bmatrix}. \tag{3.42}$$

Eq. 3.42 is a differential equation with a discontinuous right-hand side, since $\nabla_{\boldsymbol{d}^*}\varphi$ may have different directions on the sides of a switching surface between two obstacles. To analyze the solutions $\boldsymbol{X}(t)$ of Eq. 3.42, suppose that at a given time, the robot is at distance δ_i from obstacle i and distance δ_j from obstacle j. We then replace the vector field $\boldsymbol{f}^*$ in 3.42 with $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$, where $\boldsymbol{f}_i$ is the vector field on the side of the switching surface that contains obstacle i, and $\boldsymbol{f}_j$ is the vector field on the side that contains obstacle j (see Fig. 3.6):

$$\begin{aligned} \boldsymbol{f}_i(\boldsymbol{X}) &= \begin{bmatrix} \boldsymbol{X}_2 \\ -\boldsymbol{K}(\boldsymbol{X}_2 - \boldsymbol{v}_{des}) - \boldsymbol{K}_R \nabla_{\boldsymbol{d}_i} \varphi(\delta_i) \end{bmatrix}, \\ \boldsymbol{f}_j(\boldsymbol{X}) &= \begin{bmatrix} \boldsymbol{X}_2 \\ -\boldsymbol{K}(\boldsymbol{X}_2 - \boldsymbol{v}_{des}) - \boldsymbol{K}_R \nabla_{\boldsymbol{d}_j} \varphi(\delta_j) \end{bmatrix}. \end{aligned} \tag{3.43}$$

On each side of the switching surface, the robot's dynamics are described by Eq. 3.4, and therefore exhibit the desired velocity convergence and collision avoidance behaviors as we proved in Sections 3.3.1 and 3.3.2. On the switching surface, however, the closed-loop system 3.42 can have two types of solutions, depending on the directions of the vector fields $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ with respect to the switching surface. If the components of $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ that are normal to the switching surface are pointing in the same direction (Fig. 3.7, *left*), then the solution of the closed-loop system is a *Carathéodory* solution, which is an absolutely continuous function satisfying the integral equation corresponding to Eq. 3.42, $\boldsymbol{X}(t) = \boldsymbol{X}(t_0) + \int_{t_0}^{t} \boldsymbol{f}^*(\boldsymbol{X}(\tau))d\tau$ (Liberzon, 2003). In this case, the system trajectory passes through the switching surface. If the two components that are normal to the switching surface point in opposite directions (Fig. 3.7, *right*), then the system has a *Filippov* solution that satisfies the following *differential inclusion* (Liberzon, 2003), defined in terms of a convex combination of $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$:

$$\dot{\boldsymbol{X}} \in \boldsymbol{F}(\boldsymbol{X}) := \left\{ \alpha \boldsymbol{f}_i(\boldsymbol{X}) + (1-\alpha)\boldsymbol{f}_j(\boldsymbol{X}) : \alpha \in [0,1] \right\}. \tag{3.44}$$

Equation 3.44 describes the dynamics of the robot as:

$$\dot{\boldsymbol{X}} = \begin{cases} \boldsymbol{f}_i(\boldsymbol{X}), & \delta_i < \delta_j \\ \alpha \boldsymbol{f}_i(\boldsymbol{X}) + (1-\alpha)\boldsymbol{f}_j(\boldsymbol{X}), & \delta_i = \delta_j \\ \boldsymbol{f}_j(\boldsymbol{X}), & \delta_i > \delta_j \end{cases} \tag{3.45}$$

Since the components of $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ that are normal to the switching surface are pointing in opposite directions, the system trajectory corresponding to the Filippov solution can only evolve on the switching surface. At the point where the system trajectory reaches the switching surface, there is a unique convex combination of $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ (i.e., a unique value for α in 3.44) that is tangent to this surface, which defines the direction of $\boldsymbol{F}(\boldsymbol{X})$ on the surface. At each point on the switching surface, the Filippov solution is represented by the value of α for which $\boldsymbol{F}(\boldsymbol{X})$ is tangent to the surface at that point.

A trajectory corresponding to a Filippov solution often chatters about the switching surface. We note that the proposed controller, in contrast to a sliding mode controller, is not designed to stabilize the system trajectories to the switching surface. Chattering might occur for some time, but the robot will eventually leave the switching surface if certain conditions hold. Theorem 3.4.1 below guarantees that,

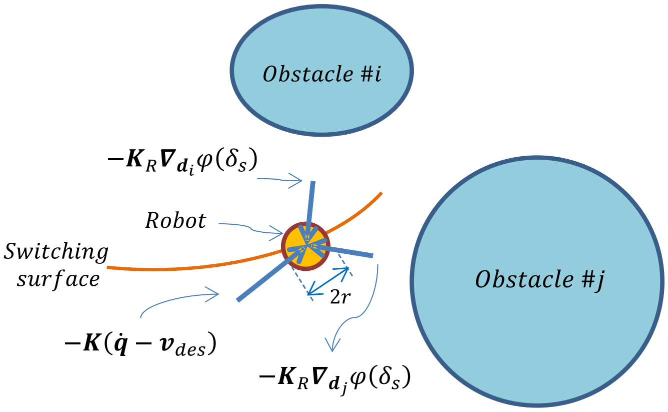

Obstacle #i
$-\boldsymbol{K}_R\boldsymbol{\nabla}_{\boldsymbol{d}_i}\varphi(\delta_s)$
Robot
Switching surface
$2r$
Obstacle #j
$-\boldsymbol{K}(\dot{\boldsymbol{q}} - \boldsymbol{v}_{des})$
$-\boldsymbol{K}_R\boldsymbol{\nabla}_{\boldsymbol{d}_j}\varphi(\delta_s)$

Figure 3.6: Illustration of the Forces That Act on the Robot When It Detects Multiple Obstacles in its Sensing Range.

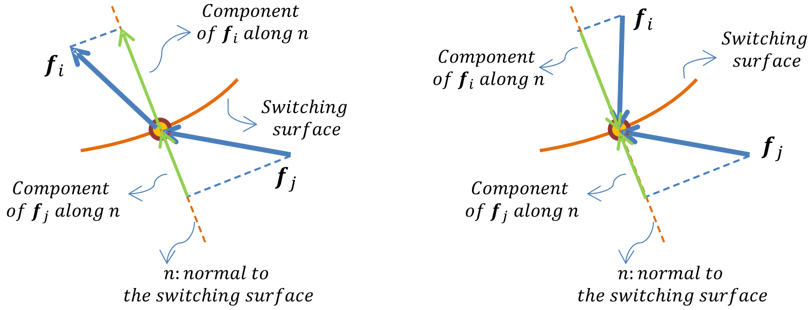

Component of $\boldsymbol{f}_i$ along n
$\boldsymbol{f}_i$
Switching surface
$\boldsymbol{f}_j$
Component of $\boldsymbol{f}_j$ along n
n: normal to the switching surface
$\boldsymbol{f}_i$
Component of $\boldsymbol{f}_i$ along n
Switching surface
$\boldsymbol{f}_j$
Component of $\boldsymbol{f}_j$ along n
n: normal to the switching surface

Figure 3.7: A Schematic Representation of Two Vector Fields That Result in (*Left*) Carathéodory and (*Right*) Filippov Solutions for a Differential Equation with a Discontinuous Right-Hand Side.

under these conditions, the closed-loop system has no equilibria on the switching surface, which ensures that the robot does not become stuck between two obstacles.

Theorem 3.4.1. *Consider an unbounded environment with at least two obstacles for which Assumption 3.1.4 holds true; i.e., the closest pair of obstacles is separated by a distance larger than the robot's diameter $2r$. Given the discontinuous control law in Eq. 3.40, no equilibrium point exists on the switching surface between any two obstacles in the environment if p in Eq. 3.1 is sufficiently small.*

Proof. Suppose that obstacles i and j are the closest pair of obstacles in an environment. By Assumption 3.1.4, the distance between these obstacles is greater than $2r$. If there exists an equilibrium point ($\dot{\boldsymbol{X}} = \boldsymbol{0}$) on the switching surface between obstacles i and j, we have that

$$\alpha \boldsymbol{f}_i(\boldsymbol{X}) + (1-\alpha)\boldsymbol{f}_j(\boldsymbol{X}) = \boldsymbol{0}. \tag{3.46}$$

Using the fact that $\delta_i = \delta_j$ on the switching surface, and substituting Eq. 3.3 for φ and Eq. 3.43 for $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$, Eq. 3.46 becomes:

$$\boldsymbol{K}\boldsymbol{v}_{des} + k_R \frac{p}{\delta_c}\left(1 - \left(\frac{\delta_c}{\delta_s}\right)^p\right)\left(\alpha \boldsymbol{e}_{\boldsymbol{d}_i} + (1-\alpha)\boldsymbol{e}_{\boldsymbol{d}_j}\right) = \boldsymbol{0}, \tag{3.47}$$

where we have defined $\delta_s := \delta_i = \delta_j$.

We now derive a conservative upper bound for the parameter p in the potential field. When the robot is on the switching surface, the repulsive force on it has the highest possible component in the direction opposite to $\boldsymbol{v}_{des}$ when $\boldsymbol{e}_{\boldsymbol{d}_i} = \boldsymbol{e}_{\boldsymbol{d}_j}$.[4] The magnitude of the repulsive force is highest when $\delta_s = r$. Substituting $\boldsymbol{e}_{\boldsymbol{d}_i} = \boldsymbol{e}_{\boldsymbol{d}_j}$ and $\delta_s = r$ into Eq. 3.47, we can reduce this equation to the following scalar equation:

$$k v_{des} + k_R \frac{p}{\delta_c}\left(1 - \left(\frac{\delta_c}{r}\right)^p\right) = 0. \tag{3.48}$$

To prevent the existence of an equilibrium point, and to ensure that the robot converges to the desired velocity, we need the stabilizing term to exceed the repulsive term; i.e.,

$$k v_{des} > -k_R \frac{p}{\delta_c}\left(1 - \left(\frac{\delta_c}{r}\right)^p\right). \tag{3.49}$$

We can rearrange this inequality to obtain the following upper bound on a function of p, called $\mu(p)$:

$$\mu(p) := p\left(\left(\frac{\delta_c}{r}\right)^p - 1\right) < \frac{k v_{des}}{k_R}\delta_c. \tag{3.50}$$

If the closest pair of obstacles are both in the robot's sensing range, we know that $r \leq \delta_c$, and therefore can confirm that $\mu(p)$ in Eq. 3.50 is strictly increasing for

[4] This is a theoretical scenario that would not happen in practice; we are using it here to obtain a conservative bound on p.

positive p. Hence, we can conclude that p must be small enough for Eq. 3.50 to hold, which completes the proof. □

The result in Theorem 3.4.1 can be generalized for a point that is equidistant from $l \in \{3, ..., m\}$ obstacles. At such a point, the convex combination of vector fields $\boldsymbol{f}_i$, which defines the differential inclusion in Eq. 3.44, is given by $\boldsymbol{F}(\boldsymbol{X}) := \sum_{i=1}^{l} \alpha_i \boldsymbol{f}_i(\boldsymbol{X})$, where $\alpha_i \in [0,1]$ for all $i \in \{1, ..., l\}$ and $\sum_{i=1}^{l} \alpha_i = 1$. The vector $\boldsymbol{X}$ is an equilibrium if $\sum_{i=1}^{l} \alpha_i \boldsymbol{f}_i(\boldsymbol{X}) = \boldsymbol{0}$, which implies that

$$\boldsymbol{K}\boldsymbol{v}_{des} + k_R \frac{p}{\delta_c}\left(1 - \left(\frac{\delta_c}{\delta_s}\right)^p\right)\left(\sum_{i=1}^{l} \alpha_i \boldsymbol{e}_{\boldsymbol{d}_i}\right) = \boldsymbol{0}, \tag{3.51}$$

where $\delta_s := \delta_1 = \delta_2 = ... = \delta_l$. Again, we consider the repulsive force on the robot with the highest possible component in the direction opposite to $\boldsymbol{v}_{des}$, which occurs when $\boldsymbol{e}_{\boldsymbol{d}_1} = \boldsymbol{e}_{\boldsymbol{d}_2} = ... = \boldsymbol{e}_{\boldsymbol{d}_l}$. This simplifies the summation in Eq. 3.51 to $\sum_{i=1}^{l} \alpha_i \ \boldsymbol{e}_{\boldsymbol{d}_i} = \boldsymbol{e}_{\boldsymbol{d}_1}$. Finally, setting $\delta_s = r$, Eq. 3.51 is simplified to Eq. 3.48. This shows that choosing p small enough to satisfy Eq. 3.50 will also guarantee the absence of an equilibrium at a point that is equidistant from three or more obstacles.

3.5 Simulation Results

To validate our controller, we simulated the motion of a disk-shaped holonomic robot in environments with a single strictly convex obstacle or multiple strictly convex obstacles. The robot's radius is $r = 0.1$ m, and its sensing radius is $\delta_c = 0.5$ m. The desired velocity is set to $v_{des} = 0.1$ m/s along the x-axis of the global frame for all the simulations. We present results for one scenario with a single obstacle and two scenarios with multiple obstacles. In all scenarios, the robot starts its motion in the safe area to the left of the obstacles, does not have any global localization or prior information about the shapes and locations of the obstacles, and only knows the desired velocity.

3.5.1 Single Obstacle

We first consider an environment with an elliptical obstacle. The control parameters in Eq. 3.2 are set to $k = 1$, $k_R = 0.05$, and $p = 0.4$. Figure 3.8 plots the trajectory of the robot in this environment, showing that it travels past the obstacle without collision. The robot's x and y velocity components over its trajectory are plotted versus time in Fig. 3.9. This figure shows that the robot quickly approaches the desired velocity until it detects the obstacle in its sensing range. Then, the controller redirects the robot so that it travels around the obstacle, as indicated by the increase in the y velocity component, and the robot deviates from the desired velocity. The robot converges to the desired velocity after it travels far enough from the obstacle that it cannot detect it within its sensing range.

3.5.2 Multiple Obstacles

In the first scenario, we consider an environment with six identical circular obstacles, shown in Fig. 3.10. The radius of each obstacle is 1.1 m, and Assumption 3.1.4 is satisfied. The control parameters are set to $k = 1$, $k_R = 0.05$, and $p = 0.32$. Fig. 3.10 shows that the robot travels between the obstacles without colliding with them or becoming entrapped. Figure 3.11 plots the time evolution of the robot's velocity components, which oscillate as the robot maneuvers between the obstacles. The sudden changes in the y velocity component occur at times when the robot detects a new obstacle in its sensing range and begins to circumvent the obstacle. The robot converges to the desired velocity after it travels past all six obstacles.

In the second scenario, we consider an environment with four different strictly convex obstacles, shown in Fig. 3.12. Assumption 3.1.4 is satisfied, since the shortest distance between the closest pair of obstacles (Obstacles 1 and 2) is 0.3 m. As illustrated in Fig. 3.12, the robot travels past the obstacles without colliding with them or becoming entrapped between them. Figure 3.13 plots the time evolution of the robot's velocity components. The robot's velocity displays a chattering behavior between times A and B, when it passes through the narrow channel between Obstacles 1 and 2. This is due to its frequent crossing of the switching surface between these two obstacles, which indicates that its trajectory is a Filippov solution of the closed-loop dynamics 3.43-3.44, as described in Section 3.4 (Fig. 3.7, *right*). At time C, the robot crosses the switching surface between Obstacles 2 and 3. No chattering occurs at this time, since the resultant of the velocity stabilizing force and the two repulsive forces from Obstacles 2 and 3 prevent the robot from entering the narrow channel between these two obstacles. At time D, the robot crosses the switching surface between Obstacles 3 and 4. No chattering is observed at this time either, because the channel between these two obstacles is relatively wide. The absence of chattering about the last two switching surfaces indicates that the robot's trajectory through these switching surfaces is a Carathéodory solution of the closed-loop dynamics, as discussed in Section 3.4 (Fig. 3.7, *left*). The robot stops sensing Obstacle 3 after it passes the corresponding point D in Fig. 3.12 and is repelled only by Obstacle 4. Fig. 3.13 shows that after circumventing all the obstacles, the robot converges to v_{des}.

3.6 Application to Velocity Control of Collective Transport in Unbounded Domains with Strictly Convex Obstacles

In this section, we propose decentralized controllers for robots without global position information or communication to perform collective transport in unbounded environments with convex obstacles.

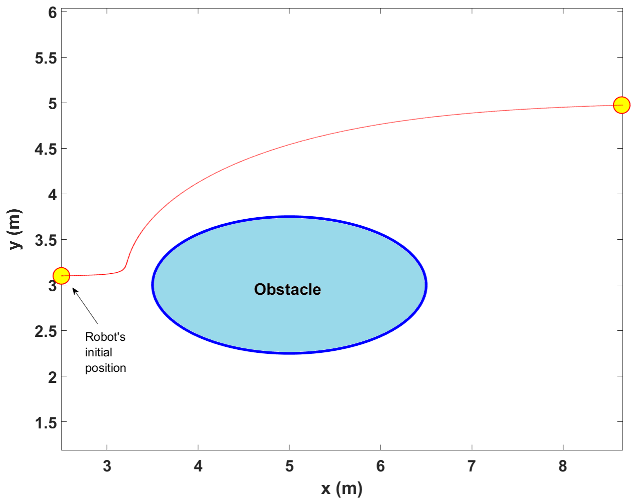


Figure 3.8: Simulation of a Disk-Shaped Holonomic Robot's Motion in an Environment with a Single Elliptical Obstacle.

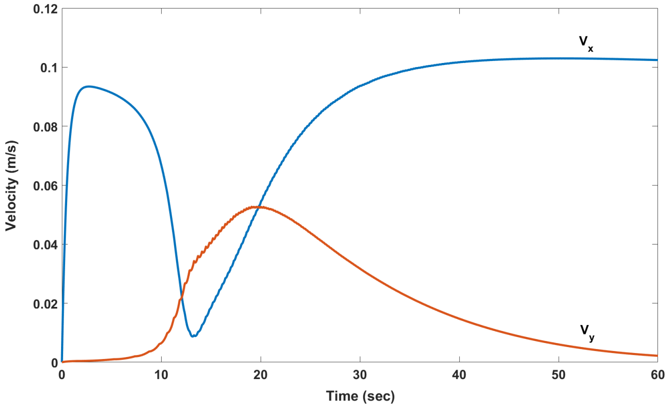


Figure 3.9: Time Evolution of the Robot's x and y Velocity Components in the Global Frame While It Moves Along the Red Trajectory Shown in Fig. 3.8.

3.6.1 Dynamical Model

We illustrate the system that we consider in Fig. 3.14. A load is transported by N point-mass robots that are rigidly attached to the load and modeled as double integrators. Here, we derive the equations of motion for the entire system, comprised of the load and the robots. We define m as the mass of the entire system and J as the moment of inertia of the entire system about the axis that passes through its center of gravity and is normal to the plane of motion. We also define $\boldsymbol{r}_i = [r_{i,x}\ r_{i,y}]^T \in \mathbb{R}^2$ as the vector from the system's center of mass (denoted by CG in

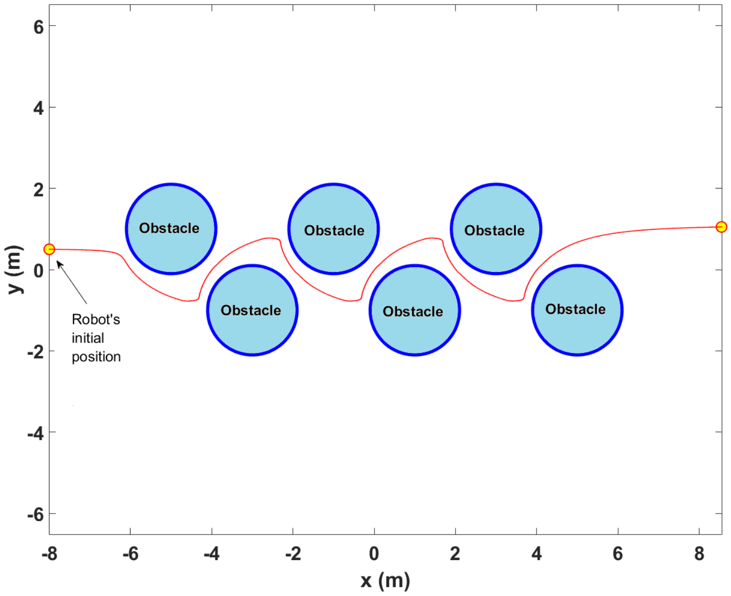


Figure 3.10: Simulation of the Robot's Motion in an Environment with Six Circular Obstacles in Which Assumption 3.1.4 Is Satisfied.

Fig. 3.14) to the attachment point of robot i. Each robot i applies an actuating force $\boldsymbol{u}_i = [u_{i,x}\ u_{i,y}]^T \in \mathbb{R}^2$ to the payload. We denote the vector of all applied forces by $\boldsymbol{u} = \left[(\boldsymbol{u}_1)^T \cdots (\boldsymbol{u}_N)^T\right]^T$. We denote the pose of the load in the inertial reference frame by $\boldsymbol{q}_o \in \mathbb{R}^3$. Then we can write the equation of motion of the system as

$$\begin{bmatrix} m\boldsymbol{I} & 0 \\ 0 & J \end{bmatrix} \ddot{\boldsymbol{q}}_o = \begin{bmatrix} \boldsymbol{I} & \cdots & \boldsymbol{I} \\ \hat{\boldsymbol{r}}_1 & \cdots & \hat{\boldsymbol{r}}_N \end{bmatrix} \boldsymbol{u}, \tag{3.52}$$

where $\boldsymbol{I} \in \mathbb{R}^{2\times 2}$ is the identity matrix and $\hat{\boldsymbol{r}}_i \in \mathbb{R}^{1\times 2}$ is defined as $\hat{\boldsymbol{r}}_i := [-r_{i,y}\ r_{i,x}]$.

3.6.2 Controller Design

The proposed decentralized controller for each robot i consists of two components:

$$\boldsymbol{u}_i = \boldsymbol{u}_{v,i} + \boldsymbol{u}_{r,i}, \tag{3.53}$$

where $\boldsymbol{u}_{v,i}$ stabilizes the robot's velocity to a target velocity $\boldsymbol{v}_{des}$, and $\boldsymbol{u}_{r,i}$ is a repulsive term that pushes the robot away from the closest obstacles in its sensing range. We

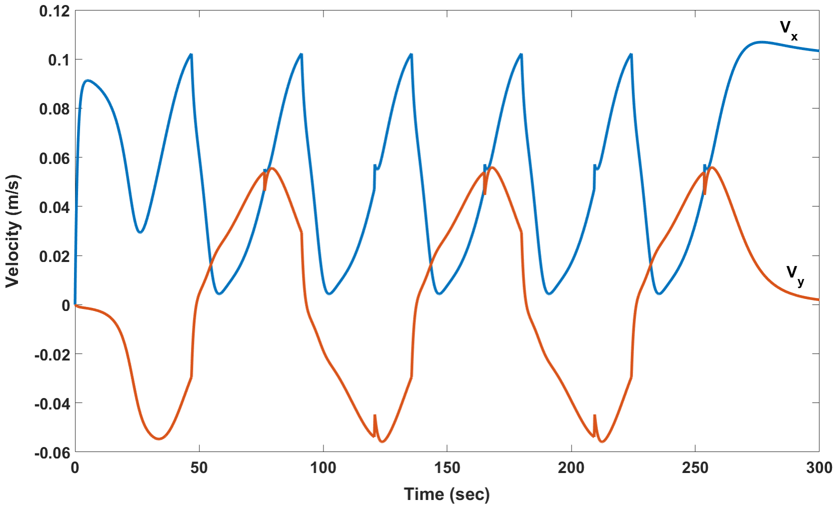


Figure 3.11: Time Evolution of the Robot's x and y Velocity Components in the Global Frame While It Moves Along the Red Trajectory Shown in Fig. 3.10.

now define each of these terms. Let $\dot{\boldsymbol{x}}_i$ be the velocity of robot i in the inertial frame. We define $\boldsymbol{u}_{v,i}$ for each robot i as a proportional-integral velocity controller:

$$\boldsymbol{u}_{v,i} = -\boldsymbol{K}(\dot{\boldsymbol{x}}_i - \boldsymbol{v}_{des}) - \boldsymbol{K}_I \int_o^t (\dot{\boldsymbol{x}}_i - \boldsymbol{v}_{des}) d\tau, \tag{3.54}$$

where $\boldsymbol{K}$ and $\boldsymbol{K}_I$ are controller gain matrices. This controller drives each robot's velocity to $\boldsymbol{v}_{des}$ and eliminates any drift from the path between the initial CG position and the goal position, as proved in (**Farivarnejad, H.** and Berman, 2018). Also, when a robot detects an obstacle in its sensing range, it applies the repulsive term in Eq. 3.40. This means that the repulsive term is written as

$$\boldsymbol{u}_{r,i} = -\boldsymbol{K}_R \boldsymbol{\nabla}_{\boldsymbol{d}^*} \varphi(\delta). \tag{3.55}$$

To this end, the decentralized control law for robot i is written as

$$\boldsymbol{u}_i = -\boldsymbol{K}(\dot{\boldsymbol{x}}_i - \boldsymbol{v}_{des}) - \boldsymbol{K}_I \int_o^t (\dot{\boldsymbol{x}}_i - \boldsymbol{v}_{des}) d\tau - \boldsymbol{K}_R \boldsymbol{\nabla}_{\boldsymbol{d}^*} \varphi(\delta). \tag{3.56}$$

We see that each robot requires only its own velocity and its distance from the closest obstacle in its sensing range to apply this controller.

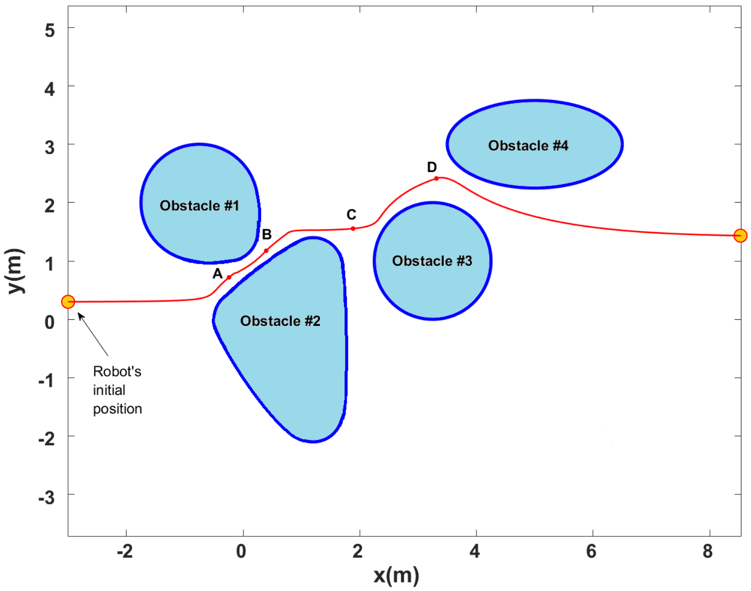


Figure 3.12: Simulation of the Robot's Motion in an Environment with Four Different Strictly Convex Obstacles in Which Assumption 3.1.4 Is Satisfied. The Points Labeled A, B, C, and D Are the Locations of the Robot at the Corresponding Times Labeled in Fig. 3.13.

3.6.3 Simulation Results

To show the effectiveness of the control law in Eq. 3.56, we consider a team of six point-mass robots that are rigidly attached to a rectangular payload. The environment contains three circular obstacles with different radii. The mass and moment of inertia of the system are $m = 1$ kg and $I = 0.33$ kg·m^2. The controller gains are set to $\boldsymbol{K} = diag(0.4, 0.4)$, $\boldsymbol{K}_I = diag(0.005, 0.005)$, and $k_r = 0.05$. The target transport speed is $v_{des} = 0.1$ m/s, and the simulation is run for 120 s. As shown in Fig. 3.15, the robots transport the payload toward the desired direction (the angle of the dashed green line) and avoid the obstacles along the way without getting stuck between them. Also, the drift from the dashed green line converges to zero after the robots pass the obstacles.

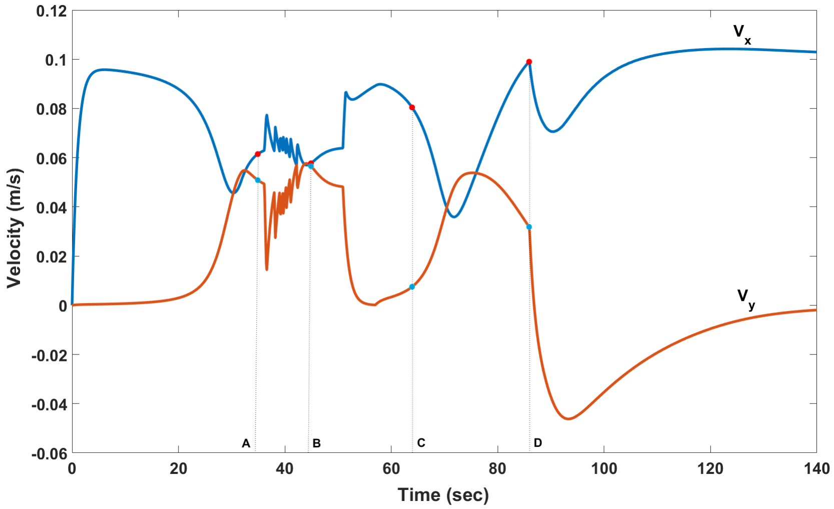


Figure 3.13: Time Evolution of the Robot's x and y Velocity Components in the Global Frame While It Moves Along the Red Trajectory Shown in Fig. 3.12.

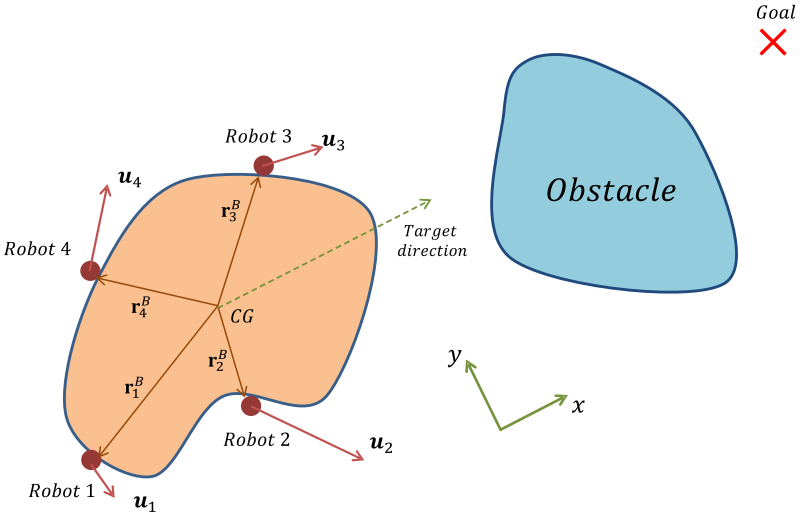


Figure 3.14: Illustration of a Collective Transport Team with Four Point-mass Robots, a Convex Obstacle, and the Associated Coordinate Systems.

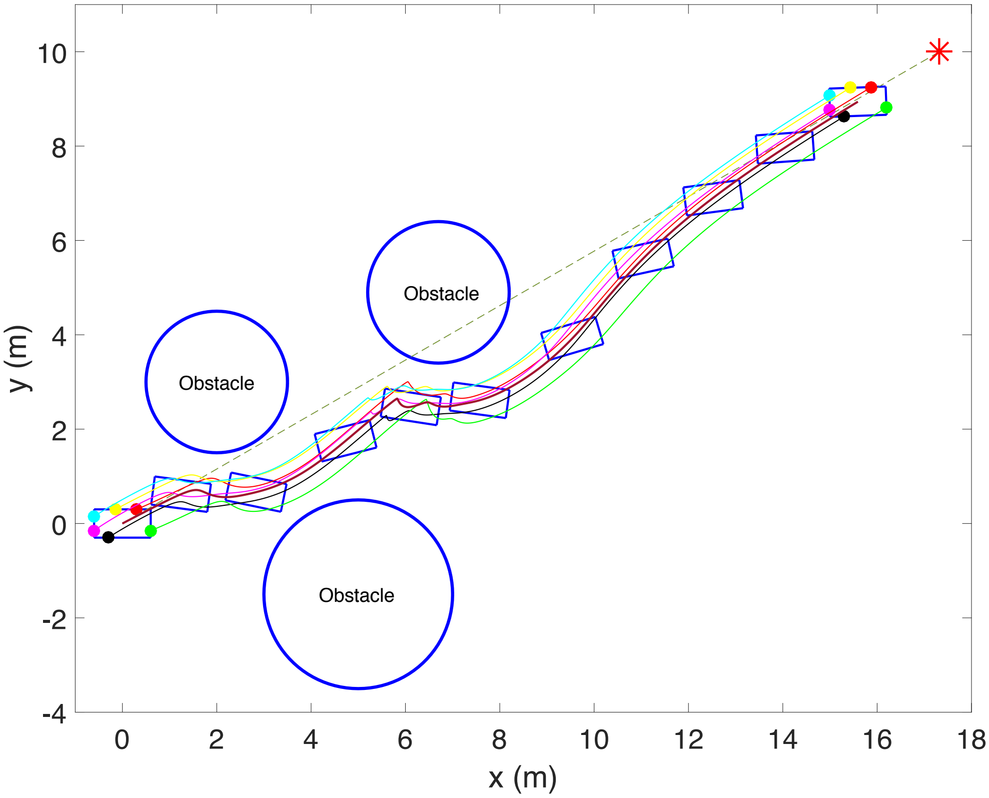


Figure 3.15: Snapshots of a Rectangular Payload Over Time with Six Robots Around Its Perimeter That Perform Collective Transport While Avoiding Locally Sensed Obstacles.

Chapter 4

POSITION CONTROL OF COLLECTIVE TRANSPORT IN UNBOUNDED DOMAINS WITHOUT OBSTACLES

In this chapter, we propose decentralized position controllers for a team of point-mass robots that must cooperatively transport a payload to a target location. The robots have double-integrator dynamics and are rigidly attached to the payload. The controllers only require robots' measurements of their own positions and velocities, and the only information provided to the robots is the desired position of the payload's center of mass. We consider scenarios in which the robots do not know the position of the payload's center of mass and try to *selfishly* stabilize their own positions to the desired location, similar to the behaviors exhibited by certain species of ants when retrieving food items in groups. We propose a proportional-derivative (PD) controller that does not rely on inter-robot communication, prior information about the load dynamics and geometry, or knowledge of the number of robots and their distribution around the payload. Using a Lyapunov argument, we prove that under this control strategy, the payload's center of mass converges to a neighborhood of the desired position. Moreover, we prove that the payload's rotation is bounded, and its angular velocity converges to zero. We show that the error between the steady-state position of the payload's center of mass and its desired position depends on the robots' distribution around the payload's center of mass, with a uniform distribution resulting in the lowest steady-state error. We validate our theoretical results with simulations in MATLAB.

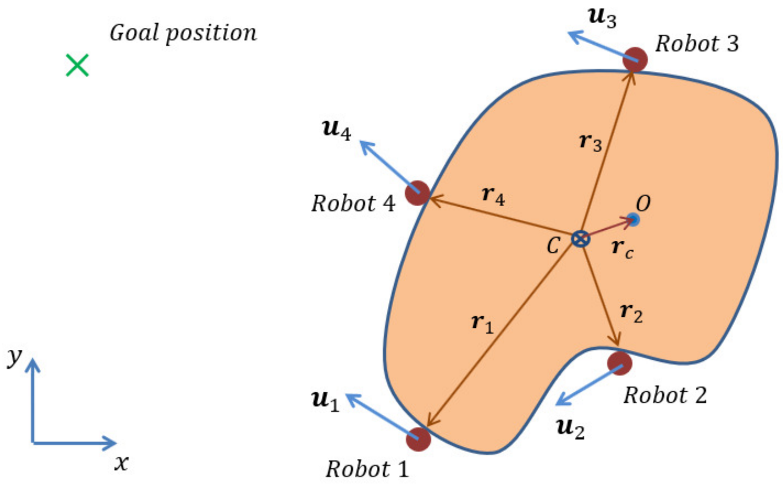


Figure 4.1: Illustration of a Collective Transport Team with Four Point-Mass Robots and the Associated Coordinate Systems.

4.1 Problem Statement

We consider a team of N identical point-mass robots that move on a planar surface and are rigidly attached to a payload in an arbitrary configuration, as shown in Fig. 4.1. We assume that each robot has access to its own position and velocity with respect to an inertial coordinate system, which is common to all the robots. The robots do not communicate with one another and are not assigned predefined trajectories. They also lack information about the payload's kinematics and dynamics, the number of robots in the transport team, and the robots' distribution around the payload.

We define $\boldsymbol{x}_o = [x_o \; y_o]^T \in \mathbb{R}^2$ and $\theta_o \in \mathbb{R}$ as the position of the payload's center of mass, point O in Fig. 4.1, and the payload's orientation with respect to a global coordinate frame, respectively. We define $\boldsymbol{x}_i = [x_i \; y_i]^T \in \mathbb{R}^2$ as the position of robot i and $\boldsymbol{x}_d = [x_d \; y_d]^T \in \mathbb{R}^2$ as the position of the target point in the global frame, as shown in Fig. 4.2. The center of mass of the entire system, including both the load and the robots, is denoted by point C in Fig. 4.1. Given that points O and C are not necessarily coincident, we define $\boldsymbol{x}_c = [x_c \; y_c]^T \in \mathbb{R}^2$ as the position of C in the

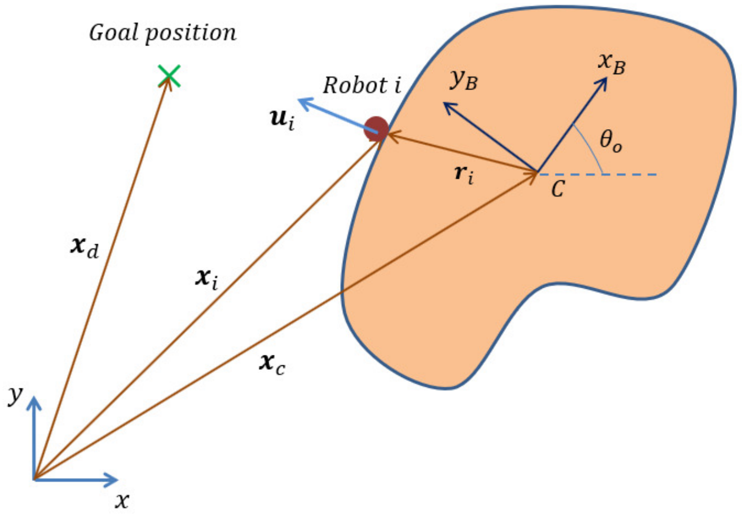


Figure 4.2: Illustration of the Geometric Parameters That Express the Position of a Robot in the Local Coordinate Frame of the Load.

global frame and $\mathbf{r}_c \in \mathbb{R}^2$ as the vector from C to O, as shown in Fig. 4.1. We also define $\boldsymbol{r}_i = [r_{ix}\ r_{iy}]^T \in \mathbb{R}^2$ as the vector from C to the attachment point of robot i in the global frame.

Each robot i knows its own position $\boldsymbol{x}_i$ and velocity $\dot{\boldsymbol{x}}_i$ and applies an actuating force $\boldsymbol{u}_i = [u_{ix}\ u_{iy}]^T \in \mathbb{R}^2$ to the payload. The control objective is to design the forces $\boldsymbol{u}_i$, $i = 1, ..., N$, such that the robots drive the position of the payload's center of mass, $\boldsymbol{x}_o$, to the target position $\boldsymbol{x}_d$. The only sensor feedback available to the robots consists of their on-board measurements of their own positions and velocities.

4.2 Dynamical Model

To derive the dynamical model of the entire system, comprised of both the load and the robots, we use the framework in Chapter 2. We denote the mass of each robot and the mass of the payload by m_r and m_o, respectively. We also define I_o as the payload's moment of inertia about the axis perpendicular to the plane and passing through O. Considering the entire system as a rigid body and defining $\boldsymbol{q} := [x_c\ y_c\ \theta_o]^T \in \mathbb{R}^3$ as the vector of generalized coordinates, we can write the equation of motion of the entire system as

$$\begin{bmatrix} m\boldsymbol{I} & 0 \\ 0 & I \end{bmatrix} \ddot{\boldsymbol{q}} = \begin{bmatrix} \boldsymbol{I} & \cdots & \boldsymbol{I} \\ \hat{\boldsymbol{r}}_1^T & \cdots & \hat{\boldsymbol{r}}_N^T \end{bmatrix} \boldsymbol{u}, \tag{4.1}$$

where m and I are the mass and moment of inertia of the entire system, given by:

$$\begin{aligned} m &= m_o + N m_r, \\ I &= I_o + m_o \left\|\mathbf{r}_c\right\|^2 + m_r \sum_{i=1}^{N} \left\|\mathbf{r}_i\right\|^2, \end{aligned} \tag{4.2}$$

and $\hat{\boldsymbol{r}}_i \in \mathbb{R}^2$ and $\boldsymbol{u} \in \mathbb{R}^{2N}$ are defined as

$$\hat{\boldsymbol{r}}_i = [-r_{iy} \quad r_{ix}]^T, \tag{4.3}$$

$$\boldsymbol{u} = [\boldsymbol{u}_1^T \ \cdots \ \boldsymbol{u}_N^T]^T. \tag{4.4}$$

The matrix $\boldsymbol{I} \in \mathbb{R}^{2\times 2}$ is the identity matrix.

4.3 Controller Design

In this section, we present decentralized robot controllers for the system described by Eq. 4.1 that produce asymptotic convergence of the payload's center of mass to a neighborhood of the desired position $\boldsymbol{x}_d$. The proposed control law has a proportional-derivative (PD) structure,

$$\boldsymbol{u}_i = -\boldsymbol{K}_d \dot{\boldsymbol{x}}_i - \boldsymbol{K}_p(\boldsymbol{x}_i - \boldsymbol{x}_d), \tag{4.5}$$

in which $\boldsymbol{K}_p = K_p\boldsymbol{I}$ and $\boldsymbol{K}_d = K_d\boldsymbol{I}$ are gain matrices, where K_p and K_d are strictly positive constants. This control law implies that each robot *selfishly* tries to stabilize its own position to the target position. Since the robots are attached to distinct points on the payload's boundary, convergence of all the robots' positions to the target position is impossible. However, by each applying the decentralized controller in Eq. 4.5, the robots produce a collective transport behavior that approximately achieves the control objective defined in Section 4.1. We analyze and discuss this behavior in the next section.

4.4 Motion Analysis

To analyze the collective behavior of the entire system of the payload and robots with the proposed controller, we first derive the dynamics of the closed-loop system and then investigate the stability and convergence properties of this system.

4.4.1 Closed-Loop Dynamics

There is a holonomic kinematic constraint between the position of robot i and the position of the system's center of mass (see Fig. 4.2), given by

$$\boldsymbol{x}_i = \boldsymbol{x}_c + \boldsymbol{r}_i. \tag{4.6}$$

Taking the time derivative of Eq. 4.6, we obtain

$$\dot{\boldsymbol{x}}_i = \dot{\boldsymbol{x}}_c + \hat{\boldsymbol{r}}_i \dot{\theta}_o, \tag{4.7}$$

where $\hat{\boldsymbol{r}}_i$ is given by Eq. 4.3. We define $\boldsymbol{e}_c := \boldsymbol{x}_c - \boldsymbol{x}_d$, where $\dot{\boldsymbol{e}}_c = \dot{\boldsymbol{x}}_c$ and $\ddot{\boldsymbol{e}}_c = \ddot{\boldsymbol{x}}_c$, since $\boldsymbol{x}_d$ is constant. Substituting the expressions for $\boldsymbol{x}_i$ and $\dot{\boldsymbol{x}}_i$ in Eqs. 4.6 and 4.7 into Eq. 4.5, we obtain

$$\boldsymbol{u}_i = -\boldsymbol{K}_d(\dot{\boldsymbol{e}}_c + \hat{\boldsymbol{r}}_i \dot{\theta}_o) - \boldsymbol{K}_p(\boldsymbol{e}_c + \boldsymbol{r}_i). \tag{4.8}$$

We now incorporate the decentralized control law for $\boldsymbol{u}_i$ in Eq. 4.8 into the dynamical model in Eq. 4.1 to derive the equation of motion of the closed-loop system as

$$\begin{aligned} \boldsymbol{M}\ddot{\boldsymbol{e}}_c &= -\boldsymbol{K}_d \sum_{i=1}^{N} (\dot{\boldsymbol{e}}_c + \hat{\boldsymbol{r}}_i \dot{\theta}_o) - \boldsymbol{K}_p \sum_{i=1}^{N} (\boldsymbol{e}_c + \boldsymbol{r}_i), \\ I\ddot{\theta}_o &= -K_d \sum_{i=1}^{N} \hat{\boldsymbol{r}}_i^T (\dot{\boldsymbol{e}}_c + \hat{\boldsymbol{r}}_i \dot{\theta}_o) - K_p \sum_{i=1}^{N} \hat{\boldsymbol{r}}_i^T (\boldsymbol{e}_c + \boldsymbol{r}_i), \end{aligned} \tag{4.9}$$

where $\boldsymbol{M} = m\boldsymbol{I}$. Taking into account the facts that $\boldsymbol{r}_i \times \boldsymbol{r}_i = \boldsymbol{0}$ and $[\boldsymbol{r}_i^T\ 0]^T \times [\boldsymbol{a}^T\ 0]^T = [0\ \ 0\ \ \hat{\boldsymbol{r}}_i^T \boldsymbol{a}]^T$, where $\boldsymbol{a}$ is an arbitrary vector in $\mathbb{R}^2$, the closed-loop system in Eq. 4.9 can be rewritten as

$$\begin{aligned} \boldsymbol{M}\ddot{\boldsymbol{e}}_c &= -N\boldsymbol{K}_d\dot{\boldsymbol{e}}_c - \boldsymbol{K}_d \sum_{i=1}^{N} \hat{\boldsymbol{r}}_i \dot{\theta}_o - N\boldsymbol{K}_p \boldsymbol{e}_c - \boldsymbol{K}_p \sum_{i=1}^{N} \boldsymbol{r}_i, \\ I\ddot{\theta}_o &= -K_d \sum_{i=1}^{N} \hat{\boldsymbol{r}}_i^T \dot{\boldsymbol{e}}_c - K_d \sum_{i=1}^{N} \|\boldsymbol{r}_i\|^2 \dot{\theta}_o - K_p \sum_{i=1}^{N} \hat{\boldsymbol{r}}_i^T \boldsymbol{e}_c. \end{aligned} \tag{4.10}$$

For notational simplicity, we define $\boldsymbol{\varrho} := \sum_{i=1}^{N} \boldsymbol{r}_i$, which implies that $\hat{\boldsymbol{\varrho}} := \sum_{i=1}^{N} \hat{\boldsymbol{r}}_i$, and $\rho := \sum_{i=1}^{N} \|\boldsymbol{r}_i\|^2$. Note that while the direction of $\boldsymbol{\varrho}$ changes with the payload's rotation, its magnitude remains unchanged since the robots are rigidly attached to the payload and C is a fixed point on the payload.

4.4.2 Convergence Analysis

The equilibrium state of the closed-loop system in Eq. 4.10 is obtained by setting $\ddot{\boldsymbol{e}}_c = \dot{\boldsymbol{e}}_c = \boldsymbol{0}$ and $\ddot{\theta}_o = \dot{\theta}_o = 0$, which results in the following equations:

$$N\boldsymbol{e}_c^* + \boldsymbol{\varrho}^* = \boldsymbol{0}, \tag{4.11}$$

$$(\hat{\boldsymbol{\varrho}}^*)^T \boldsymbol{e}_c^* = 0, \tag{4.12}$$

in which the superscript * denotes the equilibrium state. Solving Eq. 4.11 for $\boldsymbol{e}_c^*$, we obtain $\boldsymbol{e}_c^* = -\frac{1}{N}\boldsymbol{\varrho}^*$. Since $\hat{\boldsymbol{\varrho}}$ is perpendicular to $\boldsymbol{\varrho}$ by definition, this shows that Eq. 4.12 is redundant. Also, since $\boldsymbol{\varrho}$ has a constant norm, the steady-state error $\boldsymbol{e}_c^*$ has a constant magnitude. The set of equilibrium states $\mathcal{E}$ is therefore obtained as

$$\mathcal{E} = \left\{ \boldsymbol{e}_c, \dot{\boldsymbol{e}}_c \in \mathbb{R}^2, \theta_o, \dot{\theta}_o \in \mathbb{R} \ \middle|\ \boldsymbol{e}_c = -\frac{1}{N}\boldsymbol{\varrho}, \dot{\boldsymbol{e}}_c = \boldsymbol{0}, \dot{\theta}_o = 0 \right\}. \tag{4.13}$$

Note that the payload's orientation θ_o is not specified in $\mathcal{E}$, which means that $\mathcal{E}$ is a manifold in the state space and not an isolated equilibrium point. To analyze the convergence of the closed-loop system's trajectories to $\mathcal{E}$, we consider the following quadratic positive semidefinite function,

$$V = \frac{1}{2}\dot{\boldsymbol{e}}_c^T \boldsymbol{M} \dot{\boldsymbol{e}}_c + \frac{1}{2} I \dot{\theta}_o^2 + \frac{1}{2N}(N\boldsymbol{e}_c + \boldsymbol{\varrho})^T \boldsymbol{K}_p (N\boldsymbol{e}_c + \boldsymbol{\varrho}), \tag{4.14}$$

which is zero in the set $\mathcal{E}$ and positive everywhere else. The time derivative of V is calculated as

$$\begin{aligned} \dot{V} = {} & \dot{\boldsymbol{e}}_c^T \boldsymbol{M} \ddot{\boldsymbol{e}}_c + I\dot{\theta}_o \ddot{\theta}_o + \frac{1}{N}(N\boldsymbol{e}_c + \boldsymbol{\varrho})^T \boldsymbol{K}_p (N\dot{\boldsymbol{e}}_c + \dot{\boldsymbol{\varrho}}) \\ = {} & \dot{\boldsymbol{e}}_c^T(-N\boldsymbol{K}_d \dot{\boldsymbol{e}}_c - \boldsymbol{K}_d \hat{\boldsymbol{\varrho}} \dot{\theta}_o - N\boldsymbol{K}_p \boldsymbol{e}_c - \boldsymbol{K}_p \boldsymbol{\varrho}) + \dot{\theta}_o(-K_d \hat{\boldsymbol{\varrho}}^T \dot{\boldsymbol{e}}_c - K_d \rho \dot{\theta}_o - K_p \hat{\boldsymbol{\varrho}}^T \boldsymbol{e}_c) \\ & + N\boldsymbol{e}_c^T \boldsymbol{K}_p \dot{\boldsymbol{e}}_c + \boldsymbol{e}_c^T \boldsymbol{K}_p \dot{\boldsymbol{\varrho}} + \boldsymbol{\varrho}^T \boldsymbol{K}_p \dot{\boldsymbol{e}}_c + \frac{1}{N} \boldsymbol{\varrho}^T \boldsymbol{K}_p \dot{\boldsymbol{\varrho}}. \end{aligned} \tag{4.15}$$

We see that many terms in the above expression cancel out. Moreover, since we can confirm that $\dot{\boldsymbol{\varrho}} = -\hat{\boldsymbol{\varrho}}\dot{\theta}_o$ and $\boldsymbol{\varrho}^T \hat{\boldsymbol{\varrho}} = 0$, the last term in the right-hand side of Eq. 4.15 is zero. Hence, $\dot{V}$ is simplified to

$$\dot{V} = -N\dot{\boldsymbol{e}}_c^T \boldsymbol{K}_d \dot{\boldsymbol{e}}_c - NK_d \rho \dot{\theta}_o^2 - \dot{\boldsymbol{e}}_c^T \boldsymbol{K}_d \hat{\boldsymbol{\varrho}} \dot{\theta}_o - \dot{\theta}_o K_d \hat{\boldsymbol{\varrho}}^T \dot{\boldsymbol{e}}_c, \tag{4.16}$$

which can be rewritten in the following quadratic form:

$$\dot{V} = -\begin{bmatrix} \dot{\boldsymbol{e}}_c^T & \dot{\theta}_o \end{bmatrix} \underbrace{\begin{bmatrix} N\boldsymbol{K}_d & \boldsymbol{K}_d \hat{\boldsymbol{\varrho}} \\ K_d \hat{\boldsymbol{\varrho}}^T & NK_d \rho \end{bmatrix}}_{\boldsymbol{Q}} \begin{bmatrix} \dot{\boldsymbol{e}}_c \\ \dot{\theta}_o \end{bmatrix}. \tag{4.17}$$

The matrix $\boldsymbol{Q} \in \mathbb{R}^{3\times 3}$ is the same matrix $\boldsymbol{Q}$ in [(**Farivarnejad, H.** and Berman, 2018), Theorem 3.1, Eq. (12)], which we proved is positive definite. This shows that $\dot{V}$ is negative semidefinite, and henceforth V remains bounded throughout the motion of the entire system. Furthermore, invoking *LaSalle's invariant principle*, we can conclude that the trajectories of the closed-loop system in Eq. 4.10 converge to a set that is characterized by $\dot{V} \equiv 0$, for which $\dot{\boldsymbol{e}}_c \equiv \boldsymbol{0}$ and $\dot{\theta}_o \equiv 0$. This is the set $\mathcal{E}$ in Eq. 4.13. Convergence of the closed-loop system's trajectories to $\mathcal{E}$ implies that as $t \to \infty$, the center of mass of the entire system (C) converges to a neighborhood of the target position $\boldsymbol{x}_d$ and the payload's angular velocity $\dot{\theta}_o$ converges to zero. The uniform continuity of θ_o implies the convergence of θ_o to a bounded value, which depends on its initial value.

To analyze the convergence of the payload's center of mass (O) to the target position, we define $\boldsymbol{r}_{i,o}$ as the vector from point O to robot i and $\boldsymbol{\varrho}_o := \sum_{i=1}^{N} \boldsymbol{r}_{i,o}$. We also define $\boldsymbol{e}_o = \boldsymbol{x}_o - \boldsymbol{x}_d$. We can confirm that for a group of robots attached rigidly to a payload,

$$\boldsymbol{r}_c = -\frac{m_r}{m_o}\boldsymbol{\varrho} = -\frac{m_r}{m}\boldsymbol{\varrho}_o. \tag{4.18}$$

Moreover, since $\boldsymbol{x}_c = \boldsymbol{x}_o - \boldsymbol{r}_c$, we can write

$$\boldsymbol{e}_c = \boldsymbol{e}_o - \boldsymbol{r}_c. \tag{4.19}$$

Substituting Eq. 4.18 for $\boldsymbol{r}_c$ into Eq. 4.19 and then incorporating the result into Eq. 4.11, we obtain

$$\boldsymbol{e}_o^* = -\frac{1}{N}\boldsymbol{\varrho}_o^*, \tag{4.20}$$

which gives the position error of the payload's center of mass at equilibrium. Like $\boldsymbol{\varrho}$, $\boldsymbol{\varrho}_o$ has a constant magnitude, since the robots are rigidly attached to the payload and O is a fixed point on the payload. Eq. 4.20 shows that the steady-state distance between the payload's center of mass and the target position depends on the number of robots N and their distribution around the payload. This distance decreases as N is increased, and for payloads with a homogeneous mass density, it decreases as the distribution of robots around the payload's center of mass approaches a uniform distribution. For non-homogeneous payloads, this distance is reduced by allocating the robots in accordance with the payload's mass distribution; e.g., increasing the number of robots around sections of the payload with high mass density. The direction of $\boldsymbol{e}_o^*$ depends on the steady-state value of the payload orientation θ_o through $\boldsymbol{\varrho}_o^*$; the steady-state orientation depends on the initial value of θ_o, as stated earlier.

4.5 Simulation Results

We validate our analysis with simulation results for collective transport by a team of identical robots that are arranged in three different distributions around a circular payload. For each simulation, we observe the time evolution of the payload's orientation, angular velocity, and the position and velocity of its center of mass. We also study the effect of the robot distribution on the steady-state error of the payload's center of mass with respect to the target position.

The load is modeled as a homogeneous circular ring with mass $m_o = 1$ kg and moment of inertia $I_o = 0.33$ kg·m^2. Six point-mass robots, each with mass $m_r = 0.05$ kg, are rigidly attached to the load. The controller gain are $K_p = 0.8$ and $K_v = 0.3$. The payload's center of mass is initially located at $\boldsymbol{x}_o(t = 0) = [3 \ \ -1.5]^T$ m. The simulations were each run for 40 s.

Figs. 4.3-4.5 show snapshots of the payload over time for each robot distribution. The robot locations are marked as red points on the perimeter of the load in its initial and final configurations. The target position is shown as a green star at the origin, and the actual trajectory of the payload's center of mass is plotted in dashed green. The red dashed line on the load indicates its orientation. The gray circles and the orange dashed lines on them show the payload and its orientation, respectively, in intermediate states. In addition, Figs. 4.6-4.7 show the time evolution of the position and velocity of the payload's center of mass for the three distributions. Also, Fig. 4.8 plots the corresponding time evolution of the load's angular position and velocity.

In the first simulation (Fig. 4.3), the robots have an equally-spaced distribution, and the load is transported to the target position with zero steady-state error. This happens because $||\boldsymbol{\varrho}_o|| = 0$, which results in Eq. 4.20 yielding $||\boldsymbol{e}_o^*|| = 0$. The position and velocity of the payload's center of mass converge to zero quickly after around 7 s (the blue lines in Figs. 4.6 and 4.7). Also, the payload shows zero rotation and angular velocity during the entire transport (the blue lines in Fig. 4.8). In the second simulation (Fig. 4.4), the robots have a nonuniform distribution for which $||\boldsymbol{\varrho}_o|| = 1.058$ m. Using Eq. 4.20, we can obtain $||\boldsymbol{e}_o^*|| = 0.176$ m. The position and velocity of the payload's center of mass converge to their steady-state values after around 10 s (the orange lines in Figs. 4.6 and 4.7), which is a little slower than in the first simulation. In addition, the load undergoes a total rotation of approximately $\theta_o = 85°$ (the orange lines in Fig. 4.8). In the third simulation (Fig. 4.5), the robots are clustered within about a quarter of the load's perimeter. For this case, $||\boldsymbol{\varrho}_o|| = 1.477$ m, and the steady-state error has increased to $||\boldsymbol{e}_o^*|| = 0.246$ m. We also see that the system convergence to equilibrium is much slower than in the first and second simulations. The payload's position and velocity converge to their steady-state values after about 25 s (the green lines in Figs. 4.6 and 4.7). The load undergoes a large rotation of about $\theta_o = 248°$, and its angular velocity converges to zero after around 35 s (the green lines in Fig. 4.8). Thus, a highly nonuniform distribution of robots significantly affects the system's steady-state error and convergence characteristics.

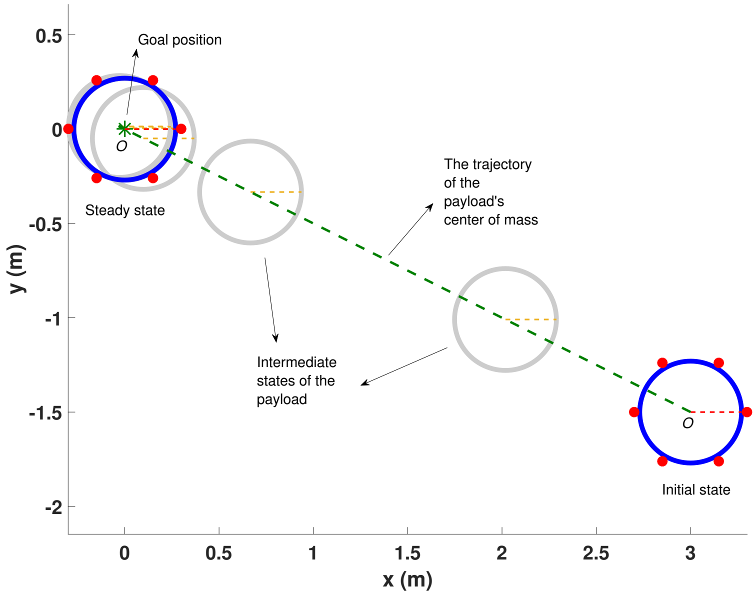


Figure 4.3: Snapshots of the Payload Over Time with an Equally-Spaced Distribution of Robots Around Its Perimeter (Distribution 1).

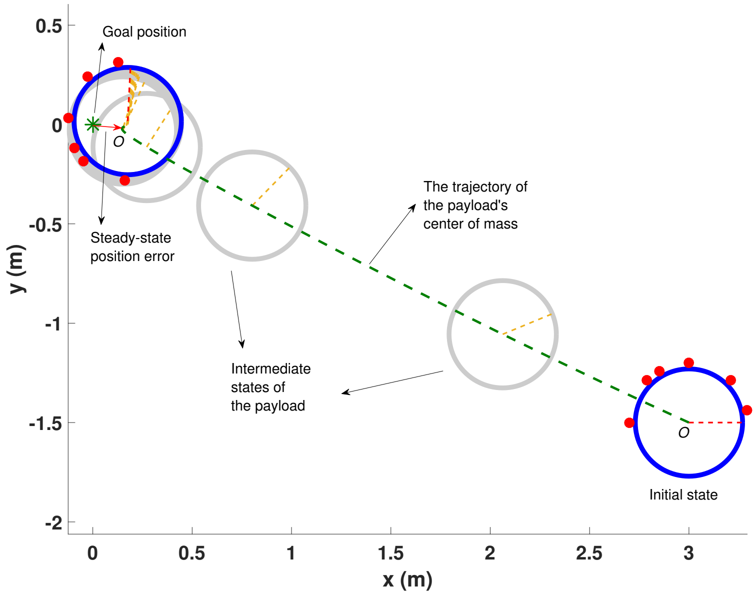


Figure 4.4: Snapshots of the Payload Over Time with a Nonuniform Distribution of Robots Around Its Perimeter (Distribution 2).

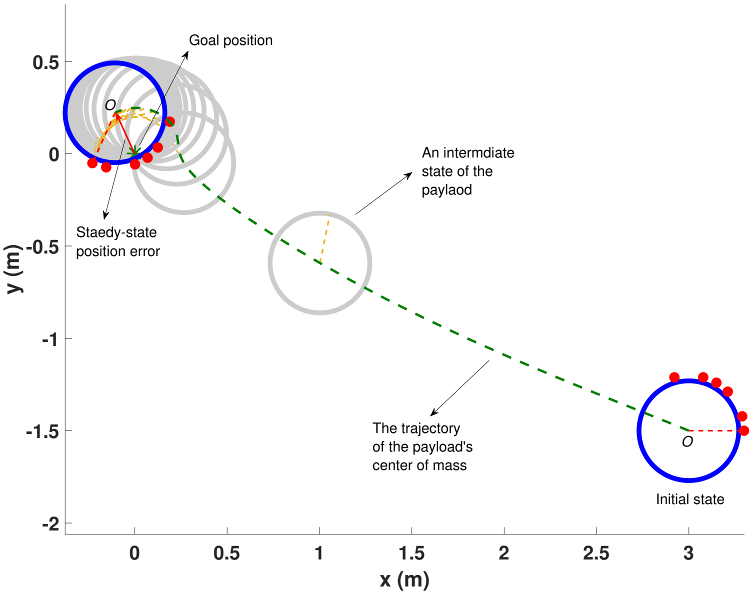
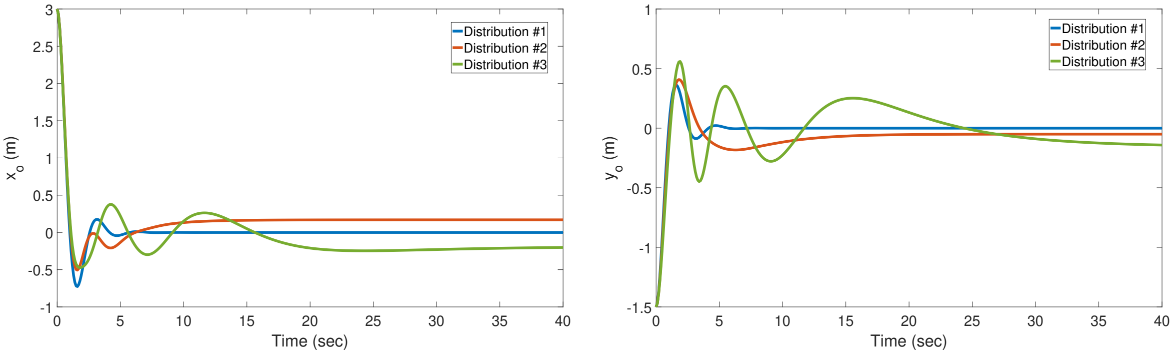


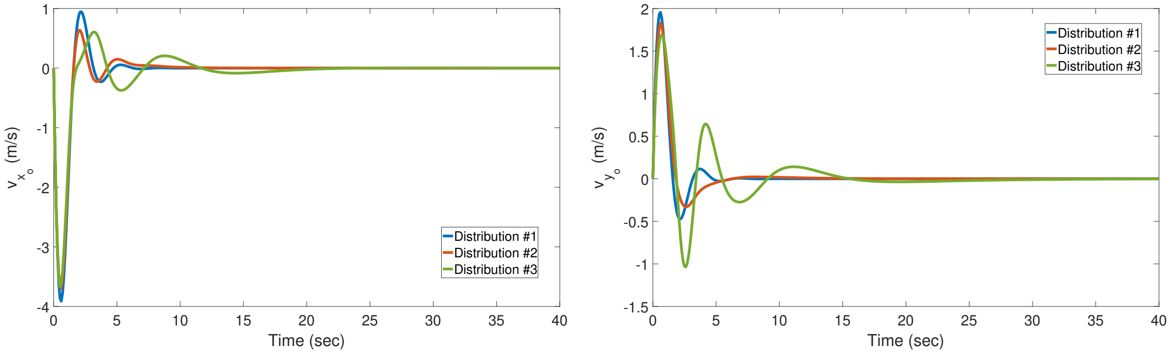
Figure 4.5: Snapshots of the Payload Over Time with a Highly Nonuniform Distribution of Robots Around Its Perimeter (Distribution 3).



(a) Position Component Along the x-axis

(b) Position Component Along the y-axis

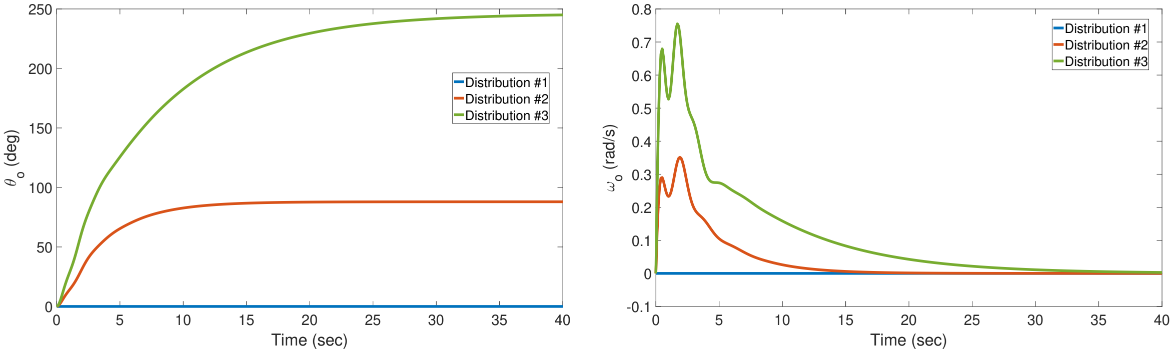
Figure 4.6: Time Evolution of the Position of the Payload's Center of Mass for the Three Distributions.



(a) Velocity Component Along the x-axis.

(b) Velocity Component Along the y-axis

Figure 4.7: Time Evolution of the Velocity of the Payload's Center of Mass for the Three Distributions.



(a) Angular Position

(b) Angular Velocity

Figure 4.8: Time Evolution of the Rotational Motion of the Payload for the Three Distributions.

Chapter 5

POSITION CONTROL OF COLLECTIVE TRANSPORT IN BOUNDED DOMAINS WITH CONVEX OBSTACLES

In this chapter, we present a controller that stabilizes a robot to a desired position and prevents it from colliding with arbitrary convex obstacles, as well as being entrapped between the obstacles, in a bounded environment. Both obstacle avoidance and convergence to the desired position are enforced by a term in the controller that is the gradient of a *navigation-like function,* which is defined in Section 5.2. As in (Arslan and Koditschek, 2019), the controller does not require the robot to have exact or approximate *a priori* information about the locations, geometries of the obstacles in the environment. However, unlike the method in (Arslan and Koditschek, 2019), we do not assume any restrictions on the curvature of the obstacles, and the proposed controller is effective for any arbitrary convex obstacle shapes. We assume that the robot has no predefined trajectory, and that it operates autonomously with minimal capabilities: it can only measure its own position, velocity, and heading and the distance to nearby obstacles within its sensing range. We also combine the proposed obstacle avoidance controller with a velocity dissipation term to design a decentralized control law for collective transport in presence of convex obstacles in a bounded convex domain.

Note: Amir Salimi Lafmejani coded the simulations in Sections 5.5 and 5.7.5 and conducted the experiments in Section 5.6.

5.1 Preliminaries and Problem Statement

We consider a disk-shape holonomic robot with radius r that moves in a planar bounded domain. The robot has first-order dynamics (a single-integrator model), $\dot{\boldsymbol{q}} = \boldsymbol{u}$, where $\boldsymbol{q} = (x, y)^T \in \mathbb{R}^2$ denotes the position of the robot in a global reference frame and $\boldsymbol{u} \in \mathbb{R}^2$ is the robot's control input. The robot can measure its distance δ from the obstacles within its sensing radius. The sensing radius is denoted by δ_c. We assume that the domain contains multiple obstacles with arbitrary convex boundaries. The control objective is for the robot to attain a desired position while avoiding collisions with the obstacles. We assume that the desired position is the origin of the global frame without loss of generality.

We first define several terms that we will frequently use throughout the paper.

Definition 5.1.1. (***Domain***). *A compact, closed and convex subset of* $\mathbb{R}^2$*, whose interior includes the origin (the target position). The domain and its boundary are denoted by* $\mathcal{D}$ *and* $\partial\mathcal{D}$*, respectively. Also, the domain's interior is denoted by* $\mathcal{I}(\mathcal{D})$ *and is defined as*

$$\mathcal{I}(\mathcal{D}) := \mathcal{D} \setminus \partial\mathcal{D}. \tag{5.1}$$

Definition 5.1.2. (***Obstacle***). *A compact, closed and convex subset of the domain, whose closure is a subset of the domain's interior and does not intersect with the domain's boundary. The domain may include multiple obstacles, which are indexed by* i*. Obstacle* i *is denoted by* $\mathcal{O}_i$*. Also, the boundary of obstacle* i *is denoted by* $\partial\mathcal{O}_i$*.*

Definition 5.1.3. (***Free Space***). *An open subset of the domain, which is obtained by removing the obstacles from the domain's interior:*

$$\mathcal{F} := \mathcal{I}(\mathcal{D}) \setminus \bigcup_{i=1}^{m} \mathcal{O}_i. \tag{5.2}$$

Definition 5.1.4. (***Repulsion Space***). *A semi-closed subset of the free space, which is bounded by an obstacle's (domain's) boundary and the closed curve that is formed by offsetting the obstacle's (domain's) boundary outward (inward) along normal to the boundary with the length δ_c. This space is denoted by $\mathcal{R}_i$ and is written as*

$$\mathcal{R}_i = \{\boldsymbol{q} \in \mathcal{F} \mid 0 < \delta_i \leq \delta_c\}, \quad i = 0, 1, \ldots, m, \tag{5.3}$$

where $\mathcal{R}_0$ represents the repulsion space associated with the domain's boundary, and the others represent the repulsion spaces of the internal obstacles.

Definition 5.1.5. (***Switching Repulsion Surface***). *If at least two repulsion spaces intersect, there exists at least one point which is at equal distances from those repulsion spaces. The set of the points that belong to the intersection of intersecting repulsion spaces and are at equal distances from their corresponding obstacles (domain) is called a switching repulsion surface. If $\mathcal{R}_j$, $\mathcal{R}_k$, $\cdots$, $\mathcal{R}_l$ form a group of intersecting repulsion spaces, the corresponding switching repulsion surface is written as*

$$\mathcal{S}_{jk\ldots l} := \left\{ \boldsymbol{q} \in \bigcap_{i=j,k,\cdots,l} \mathcal{R}_i \mid \delta_j = \delta_k = \cdots = \delta_l \right\}, \tag{5.4}$$

where δ_j, δ_k, $\cdots$, δ_l denote distances from $\mathcal{O}_j$, $\mathcal{O}_k$, $\cdots$, $\mathcal{O}_l$, respectively.

Definition 5.1.6. (***Safe Space***). *The subset of the free space that is obtained by removing the repulsion spaces from the free space is called the safe space and is defined as*

$$\mathcal{SS} := \mathcal{F} \setminus \bigcup_{i=0}^{m} \mathcal{R}_i. \tag{5.5}$$

Property 5.1.7. *The safe space has no intersection with any of the repulsion spaces:*

$$\mathcal{SS} \bigcap \mathcal{R}_i = \varnothing, \qquad \forall i \in \{0, 1, \cdots, m\}. \tag{5.6}$$

Property 5.1.8. *The safe space and the repulsion spaces form a cover of the free space:*

$$\mathcal{SS} \bigcup \left(\bigcup_{i=0}^{m} \mathcal{R}_i \right) = \mathcal{F}. \tag{5.7}$$

A circular domain with multiple circular and elliptical obstacles is shown in Fig. 5.1. Also, a schematic representation of Definitions 5.1.1 through 5.1.6 is illustrated in Fig. 5.2 for the environment in Fig. 5.1.

We make the following assumptions about the robot's capabilities. The robot has global localization and has no prior knowledge of the obstacles' locations and shapes. The robot can measure its own heading in the global frame, and it can identify the boundaries of nearby obstacles within its local sensing range, which is assumed to be a circle with radius δ_c. We assume that at each time instant, the robot can measure its distance δ, which is the length of the collision vector $\boldsymbol{d}$ according to the *Projection Theorem* in (Bertsekas *et al.*, 2003), from each obstacle within its sensing range (e.g., using infrared sensors or LIDAR). We also assume that the robot can measure the angle θ_d of the vector $-\boldsymbol{d}$ in its body-fixed frame. By adding $\theta_d + \pi$ rad to the robot's heading in the global frame, the robot can obtain the angle of the collision vector $\boldsymbol{d}$ in the global frame.

We also make the following two assumptions. The first one is about the distribution of the obstacles in the environment, which is Assumption 1 in (Arslan and Koditschek, 2019). The second assumption is about the robot's minimal knowledge about the environment.

Assumption 5.1.9. *The distance between any pair of obstacles and also the distance between each obstacle and the boundary of the domain are larger than the size of the robot, which is* $2r$.

Assumption 5.1.10. *The only information provided to the robot is the target position*

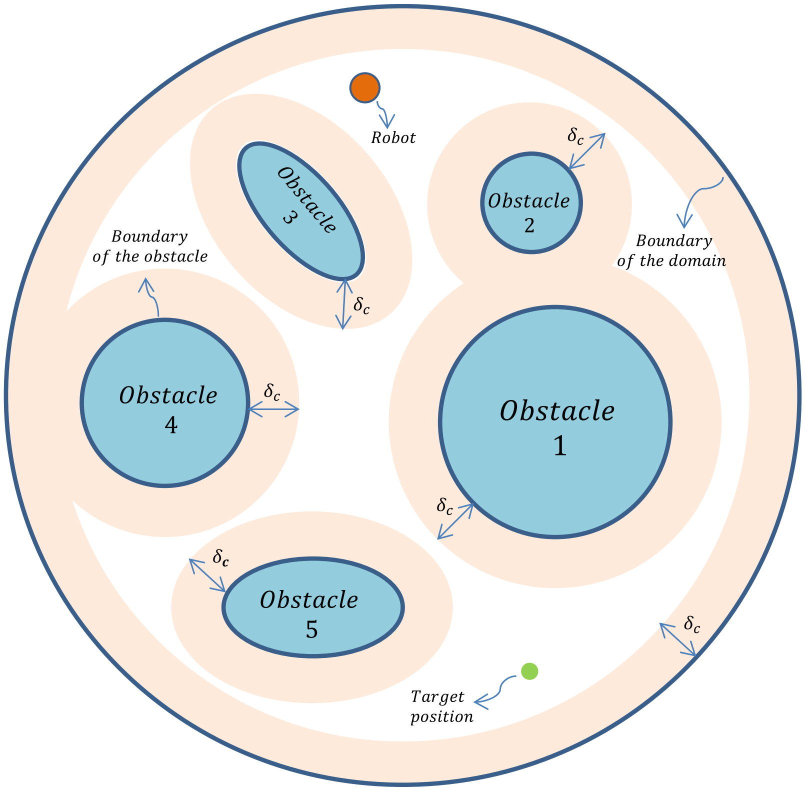


Figure 5.1: A Schematic Representation of a Circular Domain, Convex Obstacles in it, and Their Associated Areas of Effect.

and the size of the domain. The size of the domain is defined as the diameter of the smallest circle that surrounds the domain and is denoted by $2r_{\mathcal{D}}$.

Given this minimal information and completely local measurements, we seek a control law that can solve the following problem.

Problem 5.1.11. *We consider a bounded domain, whose boundary* $\partial\mathcal{D}$ *is described by* $\beta_0(x,y) = 0$*, where* $\beta_0 : \mathbb{R}^2 \mapsto \mathbb{R}$ *is a smooth function. The domain contains a finite number* $m \geq 1$ *of obstacles with arbitrary convex boundaries described by* $\beta_i(x,y) = 0$*, where each* $\beta_i : \mathbb{R}^2 \mapsto \mathbb{R}$*,* $i \in \{1, ..., m\}$ *is at least twice continuously differentiable. Given the robot's initial position in the free space* $\mathcal{F}$*, we design a robot control law that uses only the local measurements available to the robot to achieve the following objectives:*

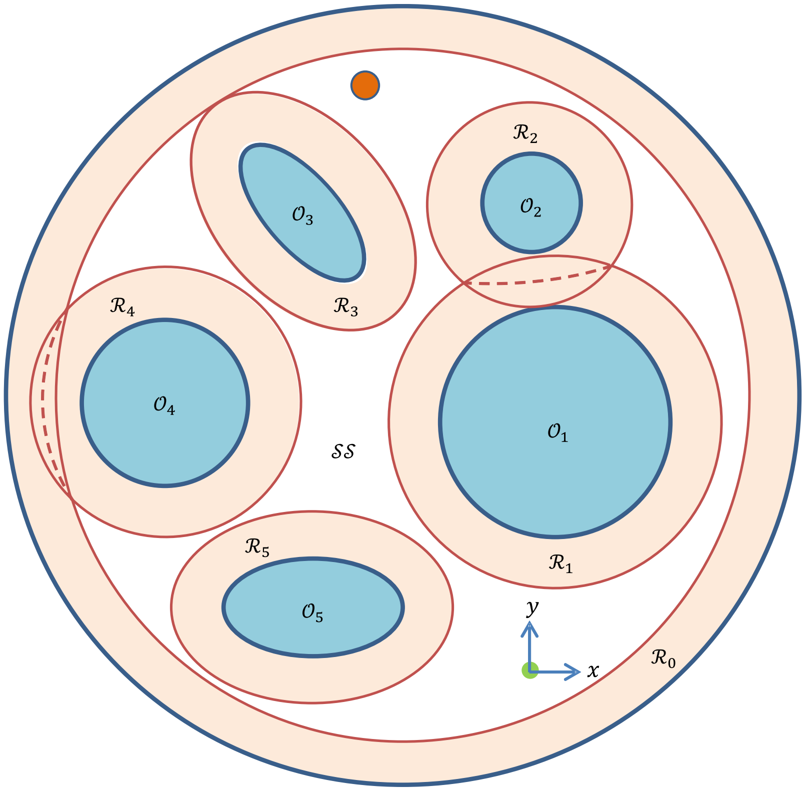


Figure 5.2: Representation of Different Spaces in the Domain Including the Safe Space, the Repulsion Spaces, and Switching Surfaces for the Configuration Illustrated in Fig. 5.1.

(1) The robot asymptotically converges to the desired position, which is inside $\mathcal{F}$ and assumed to be the origin without loss of generality.

(2) The robot does not collide with any obstacle.

(3) The robot is never trapped by any set of obstacles.

5.2 Local Navigation-Like Functions

As described in (Rimon and Koditschek, 1992), prior knowledge about the equations of the boundaries of the domain and the obstacles is required in order to construct a *navigation function* over a bounded domain. Moreover, knowledge of the number of obstacles in the domain is required to tune the parameter k in Eq. (10) in (Rimon and Koditschek, 1992) to remove all the local minima inside the free space.

This kind of information is not available in many practical applications, and this fact motivates a new definition of such functions which does not require this information about the obstacles and domain. Inspired by the original design of the navigation functions in (Rimon and Koditschek, 1992), we introduce a new notion, which we call a *navigation-like function (NLF).* These functions are defined in association with the safe space and each of the repulsion spaces. The NLF's are designed in such a way that their gradients have the following characteristics:

1) The gradients form a vector field over the entire free space $\mathcal{F}$ that steers the robot to the target position with the properties described in Problem 5.1.11.

2) The robot requires minimal information to calculate the gradients of the NLF's: specifically, the robot's position and the collision vectors (Definition 3.1.2) associated with obstacles within the robot's sensing range.

The NLF's are defined and analyzed in details in the sequel.

5.2.1 Safe Space Navigation-Like Function

The navigation-like function for the safe space $\mathcal{SS}$ is denoted by $\varphi_{\mathcal{SS}}$ and is defined as

$$\varphi_{\mathcal{SS}}(\boldsymbol{q}) = \frac{\boldsymbol{q}^T\boldsymbol{q}}{\boldsymbol{q}^T\boldsymbol{q}+1}. \tag{5.8}$$

We can confirm that $\varphi_{\mathcal{SS}} \in [0, 1)$ in a bounded domain. The next proposition characterizes the critical points of $\varphi_{\mathcal{SS}}$.

Proposition 5.2.1. *If the target position (the origin) is located in the safe space, it will be the only critical point and a global minimum for $\varphi_{\mathcal{SS}}$. Otherwise, this function has no critical point in $\mathcal{SS}$.*

Proof. The gradient of $\varphi_{\mathcal{SS}}$ is calculated as:

$$\boldsymbol{\nabla}\varphi_{\mathcal{SS}} = \frac{2\boldsymbol{q}}{(\boldsymbol{q}^T\boldsymbol{q}+1)^2}. \tag{5.9}$$

Setting $\boldsymbol{\nabla}\varphi_{\mathcal{SS}} = \mathbf{0}$, $\boldsymbol{q} = \mathbf{0}$ is the only solution provided it is inside the safe space. Also, the Hessian of $\varphi_{\mathcal{SS}}$ is written as:

$$\boldsymbol{\nabla}^2\varphi_{\mathcal{SS}} = \frac{2\left((\boldsymbol{q}^T\boldsymbol{q}+1)\boldsymbol{I} - 4\boldsymbol{q}\boldsymbol{q}^T\right)}{(\boldsymbol{q}^T\boldsymbol{q}+1)^3}, \tag{5.10}$$

where $\boldsymbol{I} \in \mathbb{R}^{2\times 2}$ is the identity matrix. The Hessian at $\boldsymbol{q} = \mathbf{0}$ is equal to $2\boldsymbol{I}$, which is a positive definite matrix. Also, from Eq. 5.8, $\varphi_{\mathcal{SS}}$ is zero at the origin and is positive every where else. Therefore, $\boldsymbol{q} = \mathbf{0}$ is a global minimum for $\varphi_{\mathcal{SS}}$. If the origin is not inside $\mathcal{SS}$, $\varphi_{\mathcal{SS}}$ has no critical point. □

Property 5.2.2. *The robot requires only its own position $\boldsymbol{q}$ to calculate the gradient of $\varphi_{\mathcal{SS}}$, as implied by Eq. 5.9.*

5.2.2 Repulsion Space Navigation-Like Function

The navigation-like function for repulsion space $\mathcal{R}_i$ is denoted by $\varphi_{\mathcal{R}_i}$ and is defined as

$$\varphi_{\mathcal{R}_i}(\boldsymbol{q}) = \frac{\boldsymbol{q}^T\boldsymbol{q}}{\boldsymbol{q}^T\boldsymbol{q} + g(\delta_i)}, \tag{5.11}$$

where $g(\delta_i) : \mathbb{R}_{>0} \mapsto \mathbb{R}_{>0}$ with $i \in \{0, 1, ..., m\}$ is a function of the robot's distance from the boundary of the domain or obstacle i and is defined as:

$$g(\delta_i) = \left(\frac{\delta_i}{\delta_c}\right)^k, \tag{5.12}$$

in which $\delta_i := \|\boldsymbol{d}_i\| - r$, δ_c is the sensing radius of the robot, and k is a strictly positive real number. We can confirm that $\varphi_{\mathcal{R}_i} \in [0, 1)$ for a bounded domain. We notice that since δ_i depends on the robot's location, $g(\delta_i)$ is implicitly a function of $\boldsymbol{q}$. The next two propositions characterize the critical points of $\varphi_{\mathcal{R}_i}$.

Proposition 5.2.3. *If the target position (the origin) is located in the repulsion space of obstacle i, it will be a global minimum for $\varphi_{\mathcal{R}_i}$.*

Proof. The gradient of $\varphi_{\mathcal{R}_i}$ is calculated as:

$$\boldsymbol{\nabla}\varphi_{\mathcal{R}_i} = \frac{2g(\delta_i)\boldsymbol{q} - (\boldsymbol{q}^T\boldsymbol{q})\boldsymbol{\nabla}g(\delta_i)}{\left(\boldsymbol{q}^T\boldsymbol{q} + g\right)^2}. \tag{5.13}$$

To find the critical points, we set $\boldsymbol{\nabla}\varphi_{\mathcal{R}_i} = \boldsymbol{0}$, which leads to the following equation:

$$2g(\delta_i)\boldsymbol{q} - (\boldsymbol{q}^T\boldsymbol{q})\boldsymbol{\nabla}g(\delta_i) = \boldsymbol{0}. \tag{5.14}$$

One solution is $\boldsymbol{q} = \boldsymbol{0}$. Also, the Hessian of $\varphi_{\mathcal{R}_i}$ is calculated as:

$$\boldsymbol{\nabla}^2\varphi_{\mathcal{R}_i} = \frac{\boldsymbol{N}_1(\boldsymbol{q}) - \boldsymbol{N}_2(\boldsymbol{q})}{(\boldsymbol{q}^T\boldsymbol{q} + g)^3}, \tag{5.15}$$

in which $\boldsymbol{N}_1(\boldsymbol{q}), \boldsymbol{N}_2(\boldsymbol{q}) \in \mathbb{R}^{2\times 2}$ are:

$$\boldsymbol{N}_1(\boldsymbol{q}) = (\rho^2 + g)\left(2g\boldsymbol{I} - \rho^2\boldsymbol{\nabla}^2 g + 2(\boldsymbol{q}\boldsymbol{\nabla}g^T - \boldsymbol{\nabla}g\boldsymbol{q}^T)\right) \tag{5.16}$$

$$\boldsymbol{N}_2(\boldsymbol{q}) = 2\left(2g\boldsymbol{q}\boldsymbol{q}^T + 2g\boldsymbol{q}\boldsymbol{\nabla}g^T - 2\rho^2\boldsymbol{\nabla}g\boldsymbol{q}^T - \rho^2\boldsymbol{\nabla}g\boldsymbol{\nabla}g^T\right), \tag{5.17}$$

where $\rho := \|\boldsymbol{q}\|$. We can see that for $\boldsymbol{q} = \boldsymbol{0}$, $\boldsymbol{N}_1 = 2g^2\boldsymbol{I}$ and $\boldsymbol{N}_2 = \boldsymbol{0}$. Thus, the Hessian at the origin is simplified to:

$$\boldsymbol{\nabla}^2\varphi_{\mathcal{R}_i}\mid_{\boldsymbol{q}=\boldsymbol{0}} = \frac{2}{g}\boldsymbol{I}, \tag{5.18}$$

which is positive definite. Furthermore, by construction, $\varphi_{\mathcal{R}_i}$ is zero at the origin and positive everywhere else in $\mathcal{R}_i$. Thus, the origin is a global minimum for $\varphi_{\mathcal{R}_i}$. □

Proposition 5.2.4. *$\varphi_{\mathcal{R}_i}$ may have non-zero norm, i.e. $\boldsymbol{q} \neq \boldsymbol{0}$, critical points in $\mathcal{R}_i$ if $k \in (0, 2)$. Also, there is no non-zero norm critical point in $\mathcal{R}_i$ for $k \in [2, \infty)$.*

Proof. The gradient of $g(\delta_i)$ in Eq. 5.14, which is originally the derivative of g with respect to $\boldsymbol{q}$, is equal to $\nabla_{\boldsymbol{d}_i}g(\delta_i)$, since $\boldsymbol{q} = \boldsymbol{d}_i + \boldsymbol{q}_{P_i}$ and g is only a function of δ_i, where $\delta_i = \|\boldsymbol{d}_i\| - r$ (similar to Equation (7) in Tanner *et al.* (2003)). Also, $\nabla_{\boldsymbol{d}_i}g(\delta_i)$ can be calculated as:

$$\boldsymbol{\nabla}_{\boldsymbol{d}_i}g(\delta_i) = g^{'}(\delta_i)\boldsymbol{e}_{\boldsymbol{d}_i}, \tag{5.19}$$

in which $g^{'}(\delta_i)$ is the derivative of g with respect to δ_i, and $\boldsymbol{e}_{\boldsymbol{d}_i}$ is the unit vector along the collision vector $\boldsymbol{d}_i$. Substituting Eq. 5.19 into Eq. 5.14 and incorporating the expressions for $g(\delta_i)$ and $g^{'}(\delta_i)$, Eq. 5.14 can be rewritten as:

$$\frac{\rho\delta_i^{k-1}}{\delta_c^k}(2\delta_i \boldsymbol{e}_{\boldsymbol{q}} - \rho k \boldsymbol{e}_{\boldsymbol{d}_i}) = \mathbf{0}, \tag{5.20}$$

where $\boldsymbol{e}_{\boldsymbol{q}}$ is the unit vectors along the robot's position vector $\boldsymbol{q}$. Therefore, the non-zero norm solutions of Eq. 5.14 are the solutions of the following equation

$$2\delta_i \boldsymbol{e}_{\boldsymbol{q}} - \rho k \boldsymbol{e}_{\boldsymbol{d}_i} = \mathbf{0}. \tag{5.21}$$

Notice that this is a vector equation, and $\boldsymbol{e}_{\boldsymbol{q}}$ and $\boldsymbol{e}_{\boldsymbol{d}_i}$ are unit vectors. Also, δ_i, ρ, and k are all positive numbers. Hence, Eq. 5.21 implies that at the non-zero norm critical points of $\varphi_{\mathcal{R}_i}$, we have:

$$\begin{aligned} \boldsymbol{e}_{\boldsymbol{q}} &= \boldsymbol{e}_{\boldsymbol{d}_i} \\ 2\delta_i &= \rho k. \end{aligned} \tag{5.22}$$

Figure 5.3 shows the origin, a convex obstacle (the ellipse), and the non-zero norm equilibrium points that are associated with the obstacle. It shows that the first equation in Eq. 5.22 can be satisfied only on the side of the obstacle that is far from the origin, since $\boldsymbol{e}_{\boldsymbol{q}}$ is always against $\boldsymbol{e}_{\boldsymbol{d}_i}$ in the side closer to the origin. This consequently implies that $\rho > \delta_i$ at these points. Defining $\rho_{P_i} := \left\|\boldsymbol{q}_{P_i}\right\|$, where $\boldsymbol{q}_{P_i}$ is the position of the projection point (Definition 3.1.1) associated with the non-zero norm critical point, and rewriting $\rho = \|\boldsymbol{d}_i\| + \rho_{P_i}$, we obtain:

$$\|\boldsymbol{d}_i\| = \frac{k}{2-k}\rho_{P_i}, \tag{5.23}$$

which gives a positive solution for $\|\boldsymbol{d}_i\|$ only for $k \in (0, 2)$. If $k \in (2, \infty)$, $\|\boldsymbol{d}_i\|$ becomes negative which means the critical point would be inside the obstacle and not in $\mathcal{R}_i$.

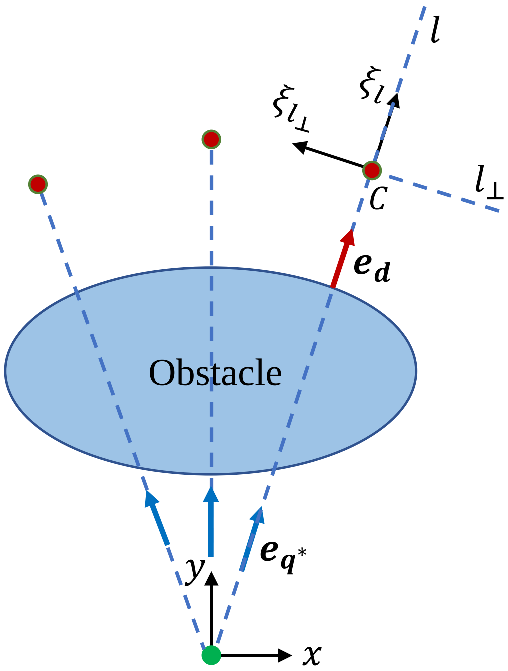


Figure 5.3: A Schematic Representation of Non-Zero Norm Equilibrium Points of a Convex Obstacle and the $\xi_l \xi_{l_\perp}$ Local Coordinate System for One of Them.

Finally, if $k = 2$, Eq. 5.22 has a solution only if the origin is located on the boundary of the obstacle, which is not the case in this problem (Item (1) in Problem 5.1.11). This completes the proof. □

There could exist one or multiple non-zero norm critical points associated with obstacle i in $\mathcal{R}_i$. The number of these points depends on the obstacle's shape and its position and orientation with respect to the global frame, as shown in Fig. 5.3. The set of all the non-zero norm critical points in $\mathcal{R}_i$ is denoted by $\mathcal{C}_i$. Also, the elements of $\mathcal{C}_i$ are denoted by $C_{ij},\ j = 1, 2, \cdots, l_i$, where l_i is the number of the non-zero norm critical points in $\mathcal{R}_i$, *i.e.*

$$\mathcal{C}_i = \{C_{ij}\}, \quad j = 1, 2, \cdots, l_i \tag{5.24}$$

The next proposition characterizes the type of these critical points.

Proposition 5.2.5. *All of the non-zero norm critical points of $\varphi_{\mathcal{R}_i}$, i.e. the elements* *of $\mathcal{C}_i$, are degenerate for $i = 0, 1, \cdots, m$.*

Proof. Incorporating Eq. 5.22 into the Hessian of $\varphi_{\mathcal{R}_i}$, *i.e.* Eq. 5.15 through Eq. 5.17, we can confirm that $\boldsymbol{N}_2 = \boldsymbol{0}$ at these points, and the Hessian is simplified to

$$\boldsymbol{\nabla}^2 \varphi_{\mathcal{R}_i} \mid_{\boldsymbol{q} \in \mathcal{C}_i} = \frac{\left(2g\boldsymbol{I} - \rho^2 \boldsymbol{\nabla}^2 g\right)}{(\boldsymbol{q}^T \boldsymbol{q} + g)^2}. \tag{5.25}$$

The Hessian of g is obtained from $\boldsymbol{\nabla}^2 g = \frac{\partial}{\partial \boldsymbol{q}} (\boldsymbol{\nabla} g)$. Taking into account Eq. 5.19, the Hessian of g can be rewritten as

$$\boldsymbol{\nabla}^2 g(\delta_i) = \frac{\partial}{\partial \boldsymbol{q}} \left(g^{'}(\delta_i) \boldsymbol{e}_{\boldsymbol{d}_i}\right) = g^{''}(\delta_i) \boldsymbol{e}_{\boldsymbol{d}_i} \boldsymbol{e}_{\boldsymbol{d}_i}^T + g^{'}(\delta_i) \left(\frac{\partial \boldsymbol{e}_{\boldsymbol{d}_i}}{\partial \boldsymbol{q}}\right). \tag{5.26}$$

Denoting the angular contribution of $\boldsymbol{e}_{\boldsymbol{d}_i}$ in the global coordinate by θ_i, and incorporating the equation for the second derivative of $g(\delta_i)$ with respect to δ_i, the first term in the right hand side of Eq. 5.26 is calculated as:

$$g^{''}(\delta_i) \boldsymbol{e}_{\boldsymbol{d}_i} \boldsymbol{e}_{\boldsymbol{d}_i}^T = \frac{k(k-1)}{\delta_c^k} \delta_i^{(k-2)} \begin{bmatrix} \cos(\theta_i)^2 & \cos(\theta_i)\sin(\theta_i) \\ \cos(\theta_i)\sin(\theta_i) & \sin(\theta_i)^2 \end{bmatrix}. \tag{5.27}$$

Furthermore, by the chain rule, the second term in the right hand side of Eq. 5.26 can be rewritten as:

$$g^{'}(\delta_i) \left(\frac{\partial \boldsymbol{e}_{\boldsymbol{d}_i}}{\partial \boldsymbol{q}}\right) = \frac{k}{\delta_c^k} \delta_i^{(k-1)} \left(\frac{\partial \boldsymbol{e}_{\boldsymbol{d}_i}}{\partial \boldsymbol{e}_{\boldsymbol{q}}}\right) \left(\frac{\partial \boldsymbol{e}_{\boldsymbol{q}}}{\partial \boldsymbol{q}}\right). \tag{5.28}$$

The vectors $\boldsymbol{e}_{\boldsymbol{d}_i}$ and $\boldsymbol{e}_{\boldsymbol{q}}$ are both unit vectors and can be related to each other by a rotation matrix as $\boldsymbol{e}_{\boldsymbol{d}_i} = \boldsymbol{R}(\alpha_i)\boldsymbol{e}_{\boldsymbol{q}}$, where α_i is the angle between the two vectors. Hence, the first parenthesis in the right hand side of Eq. 5.28 is equal to $\boldsymbol{R}(\alpha_i)$. Also, denoting the angular contribution of $\boldsymbol{e}_{\boldsymbol{q}}$ by θ, the second parenthesis is calculated as:

$$\frac{\partial \boldsymbol{e}_{\boldsymbol{q}}}{\partial \boldsymbol{q}} = \frac{1}{\rho} \begin{bmatrix} \sin(\theta)^2 & -\cos(\theta)\sin(\theta) \\ -\cos(\theta)\sin(\theta) & \cos(\theta)^2 \end{bmatrix}. \tag{5.29}$$

At the non-zero norm critical points, $\boldsymbol{e_q}$ and $\boldsymbol{e_{d_i}}$ are equal (Eq. 5.22), which implies that $\theta = \theta_i$. This results in $\boldsymbol{R}(\alpha_i) = \boldsymbol{R}(0) = \boldsymbol{I}$. Moreover, ρ in Eq. 5.29 can be replaced by $2\delta_i/k$ from Eq. 5.22. Hence, the Hessian of g can be written as:

$$\boldsymbol{\nabla}^2 g \mid_{\mathcal{C}_i} = \frac{k\delta_i^{(k-2)}}{\delta_c^k} \begin{bmatrix} (k-1)c\theta_i^2 + \frac{k}{2}s\theta_i^2 & (\frac{k}{2}-1)c\theta_i s\theta_i \\ (\frac{k}{2}-1)c\theta_i s\theta_i & (k-1)s\theta_i^2 + \frac{k}{2}c\theta_i^2 \end{bmatrix}, \tag{5.30}$$

where $c\theta_i$ and $s\theta_i$ abbreviate $\cos(\theta_i)$ and $\sin(\theta_i)$, respectively. Finally, the Hessian of $\varphi_{\mathcal{R}_i}$ at the non-zero norm critical points can be obtained from Eq. 5.25. By using Eq. 5.25, we can see that:

$$\begin{aligned} &\det\left(\boldsymbol{\nabla}^2 \varphi \mid_{\mathcal{C}_i}\right) = 0 \qquad \forall i \in \{1, 2, \cdots, m\} \\ &\operatorname{tr}\left(\boldsymbol{\nabla}^2 \varphi \mid_{\mathcal{C}_i}\right) = \frac{2-k}{2k} \qquad \forall i \in \{1, 2, \cdots, m\}. \end{aligned} \tag{5.31}$$

This demonstrates that for non-zero norm critical points inside $\mathcal{F}$, *i.e.* $k \in (0, 2)$, one eigenvalue of $\boldsymbol{\nabla}^2 \varphi_{\mathcal{R}_i}$ at the non-zero norm critical points is zero, and the other one is positive. This completes the proof. □

We conclude this section with the following properties, which are crucial in the stability analysis of the proposed controller in Section 5.4.

Property 5.2.6. *The robot requires only its own position* $\boldsymbol{q}$ *and the collision vectors* $\boldsymbol{d}_i$ *to calculate the gradient of* $\varphi_{\mathcal{R}_i}$*, as implied by Eq. 5.13 and Eq. 5.19.*

Property 5.2.7. *The repulsion space NLF's in Eq. 5.11 have the same values on switching repulsion surfaces (Definition 5.1.5), since the robot would have the same distances from the obstacles that are associated with the intersecting repulsion spaces.*

5.3 Controller Design

The control strategy for single-integrator dynamics is based on the robot's position in the free space. The proposed control law is negatively proportional to the gradient of the NLF that is associated with the position of the robot and is defined as

$$\boldsymbol{u} = -\frac{1}{n}\sum_{\eta\in\mathcal{E}} \boldsymbol{\nabla}\varphi_{\eta}(\boldsymbol{q}), \tag{5.32}$$

where

$$\mathcal{E} = \underset{\sigma\in\{\mathcal{SS},\mathcal{R}_0,\mathcal{R}_1,\cdots,\mathcal{R}_m\}}{\arg\max}\{\varphi_{\sigma}\}, \tag{5.33}$$

and $n := |\mathcal{E}|$ is the cardinality of $\mathcal{E}$. We notice that to execute the controller in Eq. 5.32 and Eq. 5.33, the robot is not required to measure its distance from every obstacle and identify the space it is located in. It instead is required only to measure and compare its distances from the obstacles within its sensing range. If the robot is in the safe space $\mathcal{SS}$, where it does not sense any obstacles in its sensing range, the controller uses the gradient of $\varphi_{\mathcal{SS}}$. When the robot is in the union of repulsion spaces and not on a switching repulsion surface, the controller uses the gradient of the $\varphi_{\mathcal{R}_i}$ that has the largest value and is essentially associated with the closest obstacle to the robot by construction (Eq. 5.11). Finally, if the robot is on a switching repulsion surface, the controller uses the average of the gradients of the NLF's of the corresponding intersecting repulsion spaces.

5.4 Analysis of Robot Motion

We incorporate the control law proposed in Eq. 5.32 and Eq. 5.33 into the robot's equation of motion ($\dot{\boldsymbol{q}} = \boldsymbol{u}$). The equation of the closed-loop system is obtained as

$$\begin{aligned}\dot{\boldsymbol{q}} &= -\frac{1}{n}\sum_{\eta\in\mathcal{E}} \boldsymbol{\nabla}\varphi_{\eta}(\boldsymbol{q}),\\ \mathcal{E} &= \underset{\sigma\in\{\mathcal{SS},\mathcal{R}_0,\mathcal{R}_1,\cdots,\mathcal{R}_m\}}{\arg\max}\{\varphi_{\sigma}\}.\end{aligned} \tag{5.34}$$

We notice that the controller proposed in Eq. 5.32 and Eq. 5.33 is a switching control law since the gradients of NLF's could discontinuously change when the robot crosses over a switching repulsion surface or passes from a repulsion space to the safe space and vice versa. This implies that the closed-loop system in Eq. 5.34 represents a switching system composed of multiple subsystems, where each subsystem is driven by the gradient of a safe space or repulsion space NLF. In this section, we analyze the switching system in Eq. 5.34 and prove that it achieves the three objectives described in Problem 5.1.11.

5.4.1 Stability Characteristics of the Equilibrium Points of the Subsystems

For each individual subsystem of the switching system in Eq. 5.34, we have $n = 1$, and henceforth the subsystem is represented as

$$\begin{aligned}
&\dot{\boldsymbol{q}} = -\boldsymbol{\nabla}\varphi_\sigma(\boldsymbol{q}), \\
&\sigma \in \{\mathcal{SS}, \mathcal{R}_0, \mathcal{R}_1, \cdots, \mathcal{R}_m\}\,. \qquad (5.35)
\end{aligned}$$

This shows that the equilibrium points of each subsystem of Eq. 5.34 are the critical points of the NLF corresponding to that subsystem. To investigate stability properties of the equilibrium points, we use Lyapunov's indirect method and linearize Eq. 5.35 around those equilibrium point as

$$\begin{aligned}
&\dot{\boldsymbol{q}} = \left(-\boldsymbol{\nabla}^2\varphi_\sigma(\boldsymbol{q}^*)\right)\boldsymbol{q}, \\
&\sigma \in \{\mathcal{SS}, \mathcal{R}_0, \mathcal{R}_1, \cdots, \mathcal{R}_m\}\,, \qquad (5.36)
\end{aligned}$$

where $\boldsymbol{q}^*$ denotes a critical point of φ_σ. Given the linearized model in Eq. 5.36, we analyze the eigenvalues of the Hessian of the corresponding NLF and use the results in Propositions 5.2.1 through 5.2.4, as discussed in the sequel.

Proposition 5.4.1. *The origin is an asymptotically stable equilibrium point if it is located either in the safe space or the union of the repulsion spaces.*

Proof. This is concluded from Propositions 5.2.1 and 5.2.3, where we proved positive definiteness of the Hessian at the origin. This consequently results in strictly negative eigenvalues for the negative of the Hessian matrix in the linearized system in Eq. 5.36. □

Non-zero norm critical points of repulsion space NLF's $\varphi_{\mathcal{R}_i}$ are other equilibrium points of the subsystems in Eq. 5.35. As proved in Proposition 5.2.4, they are degenerate with their corresponding Hessian matrices each having one zero and one positive eigenvalues. This means that the negative of the Hessian in Eq. 5.36 has one zero and one negative eigenvalue, and henceforth linearization fails to determine stability properties of these equilibrium points. To this end, the *center manifold theorem* is used to investigate stability properties of these equilibrium points (Khalil, 1996). We first state the following two lemmas, which are used in stability analysis of non-zero norm equilibria.

Lemma 5.4.2. *We define l as the line along $\boldsymbol{q}^*$ and $l_\perp$ as the line perpendicular to l, as shown in Fig. 5.3. Then $l_\perp$ and l are respectively center manifold and stable manifold for each subsystem in Eq. 5.35 at a neighborhood of their non-zero norm equilibrium points.*

Proof. We define a local coordinate system with its origin located at the equilibrium point and its axes denoted by ξ_l and $\xi_{l_\perp}$. ξ_l is along l outward the obstacle's boundary, and $\xi_{l_\perp}$ is along $l_\perp$ with its direction creating a right-handed coordinate system with ξ_l (Fig. 5.3). We also define $\boldsymbol{\xi} := [\xi_l \ \xi_{l_\perp}]^T \in \mathbb{R}^2$ as the position of a point in the local coordinate system. We know that the rotation matrix from the local coordinate to the global coordinate system equals $\boldsymbol{R}(\theta_i)$, and the transformation from the global frame to the local frame is then given by

$$\boldsymbol{\xi} = \boldsymbol{R}^T(\theta_i)(\boldsymbol{q} - \boldsymbol{q}^*). \tag{5.37}$$

We define two augmented vectors $\boldsymbol{q}_a := [\boldsymbol{q}^T \ 1]^T \in \mathbb{R}^3$ and $\boldsymbol{\xi}_a := [\boldsymbol{\xi}^T \ 1]^T \in \mathbb{R}^3$. Then Eq. 5.37 can be rewritten as

$$\boldsymbol{\xi}_a = \boldsymbol{T}\boldsymbol{q}_a \ , \quad \boldsymbol{T} = \begin{bmatrix} \boldsymbol{R}^T(\theta_i) & -\boldsymbol{R}^T(\theta_i)\boldsymbol{q}^* \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} , \tag{5.38}$$

where $\boldsymbol{T} \in \mathbb{R}^{3\times 3}$. Also, the linearized system in Eq. 5.36 can be rewritten in an augmented form in terms of $\boldsymbol{q}_a$ as

$$\begin{aligned} \dot{\boldsymbol{q}}_a &= \begin{bmatrix} -\boldsymbol{\nabla}^2\varphi_\sigma(\boldsymbol{q}^*) & \mathbf{0}_{2\times 1} \\ \mathbf{0}_{1\times 2} & 0 \end{bmatrix} \boldsymbol{q}_a, \\ \sigma &\in \{\mathcal{SS}, \mathcal{R}_0, \mathcal{R}_1, \cdots, \mathcal{R}_m\} . \end{aligned} \tag{5.39}$$

From Eq. 5.38, we have $\boldsymbol{q}_a = \boldsymbol{T}^{-1}\boldsymbol{\xi}_a$, which we use to rewrite Eq. 5.39 in terms of $\boldsymbol{\xi}_a$ as

$$\begin{aligned} \dot{\boldsymbol{\xi}}_a &= \boldsymbol{T} \begin{bmatrix} -\boldsymbol{\nabla}^2\varphi_\sigma(\boldsymbol{q}^*) & \mathbf{0}_{2\times 1} \\ \mathbf{0}_{1\times 2} & 0 \end{bmatrix} \boldsymbol{T}^{-1}\boldsymbol{\xi}_a, \\ \sigma &\in \{\mathcal{SS}, \mathcal{R}_0, \mathcal{R}_1, \cdots, \mathcal{R}_m\} . \end{aligned} \tag{5.40}$$

Doing the matrix multiplication in the right hand side of Eq. 5.40, omitting the third row and the third column in the result, which are all zeros, and removing the augmented 1 in $\boldsymbol{\xi}_a$, Eq. 5.40 is simplified to

$$\dot{\boldsymbol{\xi}} = \frac{2-k}{2k} \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix} \boldsymbol{\xi}, \tag{5.41}$$

which has the linearization matrix in a block diagonal form. This matrix has a negative eigenvalue and a zero eigenvalue for $k \in (0, 2)$. Using Theorem 8.1 in (Khalil, 1996), Eq. 5.41 demonstrates that the corresponding nonlinear system (Eq. 5.35) has a *center manifold* in the form $\xi_l = h(\xi_{l_\perp})$, where h is a smooth function. Since the vector field of the nonlinear system 5.35 (the gradient of the potential field) is along l at the equilibrium point, we can confirm that the only smooth candidate for h is the zero function, and consequently $\xi_l = 0$ is the center manifold of the nonlinear system at the equilibrium point. This shows that the $\xi_{l_\perp}$-axis of the local coordinate system is the center manifold for the nonlinear system 5.35 at the non-zero norm equilibrium point. Also, the ξ_l-axis, which is associated with the negative eigenvalue of the matrix in Eq. 5.41, is the stable manifold of the nonlinear system 5.35 at the non-zero norm equilibrium point. This completes the proof. □

Lemma 5.4.3. *We define a set $\mathcal{B}_{l_\perp}(\boldsymbol{q^*}, \epsilon)$ in the center manifold as*

$$\mathcal{B}_{l_\perp}(\boldsymbol{q^*}, \epsilon) = \{\boldsymbol{q} \in l_\perp \mid \|\boldsymbol{q} - \boldsymbol{q}^*\| \leq \epsilon,\ \epsilon > 0\}. \tag{5.42}$$

There exists $\epsilon > 0$, for which $\varphi_{\mathcal{R}_i}(\boldsymbol{q}^*)$ is maximal to every point in $\mathcal{B}_{l_\perp}(\boldsymbol{q^*}, \epsilon)$.

Proof. Eq. 5.37 yields $\boldsymbol{q}$ as

$$\boldsymbol{q} = \boldsymbol{q}^* + \boldsymbol{R}(\theta_i)\boldsymbol{\xi}. \tag{5.43}$$

We insert the expression for $\boldsymbol{q}$ in Eq. 5.43 into Eq. 5.11 and rewrite $\varphi_{\mathcal{R}_i}$ in terms of $\boldsymbol{\xi}$ as

$$\varphi_{\mathcal{R}_i} = \frac{\boldsymbol{\xi}^T\boldsymbol{\xi} + 2\boldsymbol{q}^{*^T}\boldsymbol{R}^T\boldsymbol{\xi} + \boldsymbol{q}^{*^T}\boldsymbol{q}^*}{\boldsymbol{\xi}^T\boldsymbol{\xi} + 2\boldsymbol{q}^{*^T}\boldsymbol{R}^T\boldsymbol{\xi} + \boldsymbol{q}^{*^T}\boldsymbol{q}^* + g(\delta_i)}. \tag{5.44}$$

The expression $2\boldsymbol{q}^{*^T}\boldsymbol{R}^T\boldsymbol{\xi}$ is the inner product of $\boldsymbol{q}^*$ and the representation of $\boldsymbol{\xi}$ in the global frame. These two vectors are normal to each other for any point on $l_\perp$. Hence, $\varphi_{\mathcal{R}_i}$ for any point in $\mathcal{B}_{l_\perp}$ is simplified to

$$\varphi_{\mathcal{R}_i}|_{\mathcal{B}_{l_\perp}} = \frac{\boldsymbol{\xi}^T\boldsymbol{\xi} + \boldsymbol{q}^{*^T}\boldsymbol{q}^*}{\boldsymbol{\xi}^T\boldsymbol{\xi} + \boldsymbol{q}^{*^T}\boldsymbol{q}^* + g(\delta_i)}, \qquad \forall \epsilon > 0. \tag{5.45}$$

We know that $\boldsymbol{q}^{*^T}\boldsymbol{q}^*$ is constant. Also, we can confirm that δ_i is implicitly a function of $\|\boldsymbol{\xi}\|$. Defining $\xi =: \|\boldsymbol{\xi}\|$, we can rewrite Eq. 5.45 as

$$\varphi_{\mathcal{R}_i}|_{\mathcal{B}_{l_\perp}} = \frac{\xi^2 + b}{\xi^2 + b + g(\xi)}, \qquad \forall \epsilon > 0, \tag{5.46}$$

where b is a constant that represents $\boldsymbol{q}^{*^T}\boldsymbol{q}^*$. Eq. 5.46 represents the value of $\varphi_{\mathcal{R}_i}$ for the points in $\mathcal{B}_{l_\perp}$ in terms of their distance from the equilibrium point. The derivative of $\varphi_{\mathcal{R}_i}|_{\mathcal{B}_{l_\perp}}$ with respect to ξ is calculated as

$$\frac{d}{d\xi}(\varphi_{\mathcal{R}_i}|_{\mathcal{B}_{l_\perp}}) = \frac{-g^{'}(\xi)\xi^2 + 2g(\xi)\xi - bg^{'}(\xi)}{(\xi^2 + b + g(\xi))^2}, \tag{5.47}$$

where $g^{'}(\xi)$ denotes the derivative of $g(\xi)$ with respect to ξ. We know that $g(\delta)$ is always positive and a strictly increasing function. We can also confirm that $\delta = \delta(\xi)$ is a strictly increasing function for convex obstacles. We can thus conclude that $g(\xi)$ is a strictly increasing function too, and consequently, $g^{'}(\xi)$ is strictly positive. Therefore, the numerator in the right hand side of Eq. 5.47 is a second order polynomial in terms of ξ with the coefficients being all sign-definite. To determine the sign of this polynomial, we calculate the "Δ" for the numerator as $\Delta = 4g^2 - 4bg^{'2}$ and study the following three possibilities:

1) If $\Delta < 0$, then the numerator is negative $\forall \xi > 0$ since the coefficient of ξ^2 is strictly negative.

2) If $\Delta = 0$, then the roots are obtained as $\xi_{1,2} = g/g^{'}$, and the numerator is strictly negative for $\xi \in (0, g/g^{'}) \cup (g/g^{'}, +\infty)$.

3) If $\Delta > 0$, then the roots can be written as

$$\xi_1 = \frac{g + \sqrt{g^2 - bg^{'2}}}{g^{'}}, \quad \xi_2 = \frac{g - \sqrt{g^2 - bg^{'2}}}{g^{'}}. \tag{5.48}$$

ξ_1 is positive. ξ_2 can be rewritten in the following form:

$$\xi_2 = \frac{g - (g - \zeta)}{g^{'}} = \frac{\zeta}{g^{'}}, \tag{5.49}$$

where ζ is a strictly positive variable that monotonically changes with $bg^{'2}$. Eq. 5.49 shows that ξ_2 is positive too. Since $0 < \xi_2 < \xi_1$, and the coefficient of ξ^2 is strictly negative, we can conclude that the numerator is strictly negative for $\xi \in (0, \xi_2) \cup (\xi_1, +\infty)$.

The above discussion demonstrates that the numerator in the right hand side of Eq. 5.47 is strictly negative for $\xi \in (0, \min(g/g^{'}, \zeta/g^{'}))$. Thus, the derivative is strictly negative, and $\varphi_{\mathcal{R}_i}|_{\mathcal{B}_{l_\perp}}$ is strictly decreasing for $\xi \in (0, \min(g/g^{'}, \zeta/g^{'}))$. This shows that $\varphi_{\mathcal{R}_i}|_{\mathcal{B}_{l_\perp}}(\xi = 0)$ is maximal to the points in $\xi \in (0, \min(g/g^{'}, \zeta/g^{'}))$. Since $\xi = 0 \leftrightarrow \boldsymbol{q} = \boldsymbol{q}^*$, we conclude that $\varphi_{\mathcal{R}_i}(\boldsymbol{q}^*)$ is maximal to the points in $\boldsymbol{\mathcal{B}}_{l_\perp}(\boldsymbol{q}^*, \epsilon)$ if

$$\epsilon := \min(g/g^{'}, \zeta/g^{'}). \tag{5.50}$$

This completes the proof. □

The next proposition characterizes the stability of non-zero norm equilibrium points.

Proposition 5.4.4. *The non-zero norm equilibrium points of $\varphi_{\mathcal{R}_i}$ are unstable equlibria of the subsystem in Eq. 5.35 for $i \in \{0, 1, \cdots, m\}$.*

Proof. The existence of a center manifold and a stable manifold at a non-zero norm equilibrium point (Lemma 5.4.2), and maximality of $\varphi_{\mathcal{R}_i}(\boldsymbol{q}^*)$ with respect to the points adjacent to $\boldsymbol{q}^*$ in the center manifold (Lemma 5.4.3), demonstrate that there is no neighborhood of $\boldsymbol{q}^*$ in $\mathbb{R}^2$ that encompasses $\boldsymbol{q}^*$ as a local minimum point for $\varphi_{\mathcal{R}_i}$. Thus, no basin of attraction can be introduced for $\boldsymbol{q}^*$, and it is unstable.[1] □

5.4.2 Absence of Equilibrium Points on the Switching Surfaces

The closed-loop system in Eq. 5.34 is a differential equation with a discontinuous right hand side and represents a switching system with each subsystem having the dynamics in Eq. 5.35. When the robot moves, the active subsystem can switch from

[1] Even though $\boldsymbol{q}^*$ is not technically a saddle point, it stability properties resemble those of a saddle point. It is stable for trajectories that start in a set of measure zero (l here), and it is unstable for trajectories that start outside l.

one to another based on the robot's position. This implies that the switching is *state-dependent* and creates switching surfaces in the robot's state space. When the robot evolves in a single subsystem, its dynamics are described by Eq. 5.35, with the equilibrium points analyzed in the previous section. Also, a classical solution (continuously differentiable) $\boldsymbol{q}(t)$ can be introduced for Eq. 5.34 as long as the robot evolves in a single subsystem. On the switching surface, however, the closed-loop system 5.34 can have different solutions, depending on the directions of the gradients $\boldsymbol{\nabla}\varphi_\sigma$ of the subsystems that interact on the switching surface. These solutions could form equilibrium points on a switching surface, which may possibly be stable and entrap the robot. In this section, we analyze these solution and state conditions that guarantee absence of equilibrium points on the switching surfaces. There are two types of switching surfaces for the system in Eq. 5.34. The first type is the switching repulsion surface (Definition 5.1.5). The second type is the outer margin of a repulsion space that is adjacent to the safe space. We study these two types in the sequel.

Switching repulsion surfaces

We assume that two repulsion spaces $\mathcal{R}_i$ and $\mathcal{R}_j$ intersect and consequently, a switching repulsion surface $\mathcal{S}_{ij}$ exists. This implies that the gradient $-\boldsymbol{\nabla}\varphi_{\mathcal{R}_i}$ applies to the robot on the side of the switching surface that contains obstacle i, and the gradient $-\boldsymbol{\nabla}\varphi_{\mathcal{R}_j}$ applies to the robot on the side of the switching surface that contains obstacle j. The closed-loop system 5.34 can have two types of solutions, depending on the directions of the gradients $-\boldsymbol{\nabla}\varphi_{\mathcal{R}_i}$ and $-\boldsymbol{\nabla}\varphi_{\mathcal{R}_j}$ with respect to the switching surface. If the components of $-\boldsymbol{\nabla}\varphi_{\mathcal{R}_i}$ and $-\boldsymbol{\nabla}\varphi_{\mathcal{R}_j}$ that are normal to the switching surface are pointing in the same direction, then the solution of the closed-loop system is a *Carathéodory* solution. In this case, the system trajectory passes through the switching surface, and no equilibrium point can be formed on the switching surface.

If the two components that are normal to the switching surface point in opposite directions, then the system has a *Filippov* solution that satisfies the following *differential inclusion* Liberzon (2003), defined in terms of a convex combination of $-\boldsymbol{\nabla}\varphi_{\mathcal{R}_i}$ and $-\boldsymbol{\nabla}\varphi_{\mathcal{R}_j}$:

$$\begin{aligned}&\dot{\boldsymbol{q}} \in \boldsymbol{\Upsilon}(\boldsymbol{q}) := \\ &\left\{\alpha(-\boldsymbol{\nabla}\varphi_{\mathcal{R}_i}(\boldsymbol{q})) + (1-\alpha)(-\boldsymbol{\nabla}\varphi_{\mathcal{R}_j}(\boldsymbol{q})) : \alpha \in [0,1]\right\}.\end{aligned} \tag{5.51}$$

Equation 5.51 describes the dynamics of the robot as:

$$\dot{\boldsymbol{q}} = \begin{cases} -\boldsymbol{\nabla}\varphi_{\mathcal{R}_i}(\boldsymbol{q}), & \boldsymbol{q} \in \mathcal{R}_i\ ,\quad \delta_i < \delta_j \\ \alpha(-\boldsymbol{\nabla}\varphi_{\mathcal{R}_i}(\boldsymbol{q})) + (1-\alpha)(-\boldsymbol{\nabla}\varphi_{\mathcal{R}_j}(\boldsymbol{q})), & \boldsymbol{q} \in \mathcal{S}_{ij} \\ -\boldsymbol{\nabla}\varphi_{\mathcal{R}_j}(\boldsymbol{q}), & \boldsymbol{q} \in \mathcal{R}_j\ ,\quad \delta_i > \delta_j \end{cases} \tag{5.52}$$

Since the components of $-\boldsymbol{\nabla}\varphi_{\mathcal{R}_i}$ and $-\boldsymbol{\nabla}\varphi_{\mathcal{R}_j}$ that are normal to the switching surface are pointing in opposite directions, the system trajectory corresponding to the Filippov solution can only evolve on the switching surface. At the point where the system trajectory reaches the switching surface, there is a unique convex combination of $-\boldsymbol{\nabla}\varphi_{\mathcal{R}_i}$ and $-\boldsymbol{\nabla}\varphi_{\mathcal{R}_j}$ (i.e., a unique value for α in Eq. 5.51) that is tangent to this surface, which defines the direction of $\boldsymbol{\Upsilon}(\boldsymbol{q})$ on the surface. At each point on the switching surface, the Filippov solution is represented by the value of α for which $\boldsymbol{\Upsilon}(\boldsymbol{q})$ is tangent to the surface at that point.

A trajectory corresponding to a Filippov solution often chatters about the switching surface. We note that the proposed controller, in contrast to a sliding mode controller, is not designed to stabilize the system trajectories to the switching surface. Chattering might occur for some time, but the robot will eventually leave the switching surface if certain conditions hold. Proposition 5.4.5 below guarantees that, under these conditions, the closed-loop system has no equilibria on the switching surface, which ensures that the robot does not become stuck between two obstacles.

Proposition 5.4.5. *Given Assumption 5.1.9 and Assumption 5.1.10, no equilibrium point exists on the repulsion switching surface $\mathcal{S}_{ij}$ between two intersecting repulsion spaces $\mathcal{R}_i$ and $\mathcal{R}_j$ if k in Eq. 5.12 is chosen to satisfy $k < r/r_{\mathcal{D}}$.*

Proof. By Assumption 5.1.9, the distance between obstacle i and obstacle j is greater than $2r$. If there exists an equilibrium point ($\dot{\boldsymbol{q}} = \mathbf{0}$) on the switching repulsion surface $\mathcal{S}_{ij}$, from Eq. 5.52, we have that

$$\alpha(-\boldsymbol{\nabla}\varphi_{\mathcal{R}_i}(\boldsymbol{q})) + (1-\alpha)(-\boldsymbol{\nabla}\varphi_{\mathcal{R}_j}(\boldsymbol{q})) = \mathbf{0}. \tag{5.53}$$

Using the fact that $\delta_i = \delta_j$ on the switching surface, and writing the expressions for $-\boldsymbol{\nabla}\varphi_{\mathcal{R}_i}$ and $-\boldsymbol{\nabla}\varphi_{\mathcal{R}_j}$ using Eq. 5.13 and Eq. 5.19, Eq. 5.53 becomes:

$$2g(\delta_s)\boldsymbol{e}_{\boldsymbol{q}} - \rho g^{'}(\delta_s)\left(\alpha\boldsymbol{e}_{\boldsymbol{d}_i} + (1-\alpha)\boldsymbol{e}_{\boldsymbol{d}_j}\right) = \mathbf{0}, \tag{5.54}$$

where we have defined $\delta_s := \delta_i = \delta_j$.

We now derive a conservative upper bound for the parameter k in $\varphi_{\mathcal{R}_i}$ (Eq. 5.11 and Eq. 5.12). When the robot is on the switching repulsion surface, the repulsive force on it has the highest possible component in the direction opposite to $\boldsymbol{e}_{\boldsymbol{q}}$ when $\boldsymbol{e}_{\boldsymbol{d}_i} = \boldsymbol{e}_{\boldsymbol{d}_j}$.[2] Substituting $\boldsymbol{e}_{\boldsymbol{d}_i} = \boldsymbol{e}_{\boldsymbol{d}_j}$ and $\delta_s = r$ into Eq. 5.54, we can reduce this equation to the following scalar equation:

$$2g(\delta_s) - \rho g^{'}(\delta_s) = 0. \tag{5.55}$$

To prevent the existence of an equilibrium point, and to ensure that the robot converges to the origin (the target position), we need the attraction term to exceed the repulsive term; i.e.,

$$2g(\delta_s) > \rho g^{'}(\delta_s). \tag{5.56}$$

[2] This is a theoretical scenario that would not happen in practice; we are using it here to obtain a conservative bound on k.

Using Eq. 5.12, this inequality is simplified to

$$2\delta_s > k\rho \tag{5.57}$$

Assumption 5.1.9 gives the radius of robot r is the lowest possible value for δ_s, and Assumption 5.1.10 gives the size of the domain $2r_{\mathcal{D}}$ is the highest possible value for ρ. Considering a worst case scenario, the inequality in Eq. 5.57 is rewritten as $2r > 2kr_{\mathcal{D}}$, which gives the following conservative upper bound for k:

$$k < \frac{r}{r_{\mathcal{D}}}. \tag{5.58}$$

This completes the proof. □

The result in Proposition 5.4.5 can be generalized for the repulsion switching surface of more than two obstacles, as stated in next corollary.

Corollary 5.4.6. *Consider a switching repulsion surface that lies within the intersection of $l \in \{3, ..., m\}$ repulsion spaces. We can confirm that such a switching repulsion surface is only a single point. The condition in Eq. 5.58 ensures that no equilibrium is formed at such a point.*

Proof. The convex combination of vector fields $-\boldsymbol{\nabla}\varphi_{\mathcal{R}_i}(\boldsymbol{q})$, which defines the differential inclusion in Eq. 5.51, is given by $\boldsymbol{\Upsilon}(\boldsymbol{q}) := \sum_{i=1}^{l} -\alpha_i \boldsymbol{\nabla}\varphi_{\mathcal{R}_i}(\boldsymbol{q})$, where $\alpha_i \in [0,1]$ for all $i \in \{1, ..., l\}$ and $\sum_{i=1}^{l} \alpha_i = 1$. The vector $\boldsymbol{q}$ is an equilibrium if $\sum_{i=1}^{l} \alpha_i \boldsymbol{\nabla}\varphi_{\mathcal{R}_i}(\boldsymbol{q}) = \boldsymbol{0}$, which implies that

$$2g(\delta_s)\boldsymbol{e_q} - \rho g^{'}(\delta_s)\left(\sum_{i=1}^{l} \alpha_i \boldsymbol{e_{d_i}}\right) = \boldsymbol{0}, \tag{5.59}$$

where $\delta_s := \delta_1 = \delta_2 = ... = \delta_l$. Again, we consider the repulsive force on the robot with the highest possible component in the direction opposite to $\boldsymbol{e_q}$, which occurs when $\boldsymbol{e_{d_1}} = \boldsymbol{e_{d_2}} = ... = \boldsymbol{e_{d_l}}$.This simplifies the summation in Eq. 5.59 to $\sum_{i=1}^{l} \alpha_i\ \boldsymbol{e_{d_i}} = \boldsymbol{e_{d_1}}$.

Finally, setting $\delta_s = r$, Eq. 5.59 is simplified to Eq. 5.55. This shows that choosing k small enough to satisfy Eq. 5.58 will also guarantee the absence of an equilibrium at a switching repulsion surface that is equidistant from three or more obstacles. □

Switching surfaces between the safe space and repulsion spaces

We assume that a segment of the outer margin of repulsion space $\mathcal{R}_i$ is adjacent to the safe space $\mathcal{SS}$. This then means that the gradient $-\boldsymbol{\nabla}\varphi_{\mathcal{R}_i}$ applies to the robot on the side of the switching surface that contains obstacle i, and the gradient $-\boldsymbol{\nabla}\varphi_{\mathcal{SS}}$ applies to the robot on the other side of the switching surface, which is $\mathcal{SS}$. The discussion in Subsection 5.4.2 about the discontinuity of the right hand side of Eq. 5.34 and the two possible types of solutions to Eq. 5.34 (Carathéodory and Filippov solutions) hold true here too. To this end, like Subsection 5.4.2, we consider a Filippov solution to Eq. 5.34 on the switching surface between repulsion space $\mathcal{R}_i$ and the safe space $\mathcal{SS}$, and we state the next proposition which guarantees the absence of an equilibrium point on this switching surface.

Proposition 5.4.7. *Given Assumption 5.1.10 and δ_c as the robot's sensing radius, no equilibrium point exists on the switching surface between repulsion space $\mathcal{R}_i$ and the safe space $\mathcal{SS}$ if k in Eq. 5.12 is chosen to satisfy $k < \delta_c/r_{\mathcal{D}}$.*

Proof. Taking the same procedure as in the proof of Proposition 5.4.5, an equilibrium point ($\dot{\boldsymbol{q}} = \boldsymbol{0}$) exists on the switching surface if

$$\alpha(-\boldsymbol{\nabla}\varphi_{\mathcal{SS}}(\boldsymbol{q})) + (1-\alpha)(-\boldsymbol{\nabla}\varphi_{\mathcal{R}_i}(\boldsymbol{q})) = \boldsymbol{0}. \tag{5.60}$$

Using the fact that $\delta_i = \delta_c$ and consequently, $g(\delta_i) = 1$ and $g^{'}(\delta_i) = k/\delta_c$ on the outer margin of $\mathcal{R}_i$, and writing the expressions for $-\boldsymbol{\nabla}\varphi_{\mathcal{SS}}$ and $-\boldsymbol{\nabla}\varphi_{\mathcal{R}_i}$ using Eq. 5.9, Eq. 5.13, and Eq. 5.19, Eq. 5.60 becomes:

$$2\boldsymbol{e}_{\boldsymbol{q}} - (1-\alpha)(k/\delta_c)\rho\boldsymbol{e}_{\boldsymbol{d}_i} = \boldsymbol{0}. \tag{5.61}$$

To prevent the existence of an equilibrium point, and to guarantee that the robot converges to the origin (the target position), we again set the attraction term to exceed the repulsive term; i.e.,

$$2 > (1-\alpha)(k/\delta_c)\rho. \tag{5.62}$$

The right hand side of the inequality in Eq. 5.62 has the highest possible value if $\alpha = 0$ and $\rho = 2r_{\mathcal{D}}$. Inserting these values into the right hand side of Eq. 5.62, we obtain the following conservative upper bound for k:

$$k < \frac{\delta_c}{r_{\mathcal{D}}}, \tag{5.63}$$

which guarantees absence of an equilibrium point on the switching surface between $\mathcal{R}_i$ and $\mathcal{SS}$. This completes the proof. □

Corollary 5.4.8. *Consider a switching surface that lies within the intersection of the safe space and $l \in \{2, ..., m\}$ repulsion spaces. We can confirm that such a switching surface is only a single point. The condition in Eq. 5.63 ensures that no equilibrium is formed at such a point.*

Proof. We take the same strategy as in the proof of Corollary 5.4.6. The convex combination of vector fields $-\boldsymbol{\nabla}\varphi_{\mathcal{SS}}(\boldsymbol{q})$ and $-\boldsymbol{\nabla}\varphi_{\mathcal{R}_i}(\boldsymbol{q})$, which defines the differential inclusion in Eq. 5.51, is given by $\boldsymbol{\Upsilon}(\boldsymbol{q}) := -\alpha_{ss}\boldsymbol{\nabla}\varphi_{\mathcal{SS}}(\boldsymbol{q}) - \sum_{i=1}^{l}\alpha_i\boldsymbol{\nabla}\varphi_{\mathcal{R}_i}(\boldsymbol{q})$, where $\alpha_{ss}, \alpha_i \in [0,1]$ for all $i \in \{1, ..., l\}$ and $\alpha_{ss} + \sum_{i=1}^{l}\alpha_i = 1$. The vector $\boldsymbol{q}$ is an equilibrium if $\alpha_{ss}\boldsymbol{\nabla}\varphi_{\mathcal{SS}}(\boldsymbol{q}) + \sum_{i=1}^{l}\alpha_i\boldsymbol{\nabla}\varphi_{\mathcal{R}_i}(\boldsymbol{q}) = \boldsymbol{0}$, which implies that

$$2g(\delta_s)\boldsymbol{e_q} - \rho g^{'}(\delta_s)\left(\sum_{i=1}^{l}\alpha_i\boldsymbol{e}_{\boldsymbol{d}_i}\right) = \boldsymbol{0}, \tag{5.64}$$

We know that $\delta_s := \delta_1 = \delta_2 = ... = \delta_l = \delta_c$ at this switching surface. Thus, Eq. 5.64 is reduced to

$$2\boldsymbol{e_q} - \rho(k/\delta_c)\left(\sum_{i=1}^{l}\alpha_i\boldsymbol{e}_{\boldsymbol{d}_i}\right) = \boldsymbol{0}. \tag{5.65}$$

Considering the the highest possible repulsive force in the direction opposite to $\boldsymbol{e_q}$, which occurs when $\boldsymbol{e_{d_1}} = \boldsymbol{e_{d_2}} = ... = \boldsymbol{e_{d_l}}$, the summation in Eq. 5.65 is simplified to $\sum_{i=1}^{l} \alpha_i \; \boldsymbol{e_{d_i}} = (1 - \alpha_{ss})\boldsymbol{e_{d_1}}$. Finally, setting the attraction term higher than the repulsion term to prevent the formation of an equilibrium point, we obtain

$$2 > (1 - \alpha_{ss})(k/\delta_c)\rho, \tag{5.66}$$

which resembles the inequality in Eq. 5.62 and gives the same upper bound for k as in Eq. 5.63. □

We conclude this subsection with the following theorem that guarantees the absence of equilibrium points on any switching surface in the domain.

Theorem 5.4.9. *Given Assumption 5.1.9 and Assumption 5.1.10, no equilibrium point exists on any switching surface in the domain if k in Eq. 5.12 satisfies the following condition:*

$$k < \frac{1}{r_{\mathcal{D}}} \min(r, \delta_c). \tag{5.67}$$

Proof. The result is immediately deduced from Corollary 5.4.6 and Corollary 5.4.8. □

5.4.3 Convergence Analysis

As discussed in Subsection 5.4.2, the closed-loop system in Eq. 5.34 is a differential equation with a discontinuous right hand side, which does not meet the Lipchitz continuity condition. This implies that we cannot directly apply Lyapunov and LaSalle's theorems to analyze stability/convergence properties of the closed-loop system. To this end, we use the idea of *multiple Lyapunov functions* that has been developed for stability analysis of switching systems (Chapter 3 in (Liberzon, 2003)). To this end, we first state the following two lemmas, which are used in the analysis afterward.

Lemma 5.4.10. *Given the closed-loop system in Eq. 5.34 that is composed of the subsystems in Eq. 5.35, $V_\sigma(\boldsymbol{q}) := \varphi_\sigma(\boldsymbol{q})$ is a continuous function for every solution $\boldsymbol{q}(t)$ of Eq. 5.34 for $t \geq 0$.*

Proof. The safe space NLF $\varphi_{SS}(\boldsymbol{q})$ in Eq. 5.8 and the repulsion space NLF $\varphi_{\mathcal{R}_i}(\boldsymbol{q})$ in Eq. 5.11 are continuous functions by construction. Also, when the robot crosses a switching repulsion surface between $l \in \{2, ..., m\}$ repulsion spaces, $\varphi_{\mathcal{R}_i}(\boldsymbol{q})$ does not change value since δ_i has the same value for all $i \in \{1, ..., l\}$, and consequently, $g(\delta_i)$ is the same for all the intersecting repulsion spaces on the switching repulsion surface. Finally, when the robot crosses the outer margin of repulsion space $\mathcal{R}_i$ to the safe space $\mathcal{SS}$, the value of V_σ does not change since $\delta_i = \delta_c$, and consequently, $g(\delta_i) = 1$ on the outer margin of $\mathcal{R}_i$. This means that $\varphi_{\mathcal{R}_i}(\boldsymbol{q}) = \varphi_{SS}(\boldsymbol{q})$ on the outer margin of $\mathcal{R}_i$. The same analysis holds true when the robot leaves the safe space to a repulsion space. This completes the proof. □

Lemma 5.4.11. *Considering the subsystems in Eq. 5.35, we define $t_{\sigma,i}$ as the time instant that the robot enters σ and $t_{\sigma,i+1}$ as the time instant that the robot leaves σ to another subsystem, i.e. $\boldsymbol{q}(t) \in \sigma$, $\forall t \in [t_{\sigma,i}, t_{\sigma,i+1})$. Then the function V_σ strictly decreases over the time interval $[t_{\sigma,i}, t_{\sigma,i+1})$ for every subsystem σ in Eq. 5.35. Moreover, the system trajectory converges to the origin (the target position) if it is located in σ. Finally, if there is a non-zero norm equilibrium point $\boldsymbol{q}^*$ in σ, and a trajectory starts inside the stable manifold l of this point, i.e. $t_{\sigma,i} = 0$ and $\boldsymbol{q}(0) \in l$, then the trajectory converges to $\boldsymbol{q}^*$.*

Proof. The time derivative of V_σ for $t \in [t_{\sigma,i}, t_{\sigma,i+1})$ is calculated as

$$\begin{aligned} \dot{V}_\sigma(t) &= \dot{\varphi}_\sigma(t) \\ &= \left(\boldsymbol{\nabla}\varphi_\sigma(\boldsymbol{q}(t))\right)^T \dot{\boldsymbol{q}}(t), \quad t \in [t_{\sigma,i}, t_{\sigma,i+1}). \end{aligned} \tag{5.68}$$

Inserting $\dot{\boldsymbol{q}}$ from Eq. 5.35, we obtain

$$\dot{V}_\sigma(t) = -\left\|\boldsymbol{\nabla}\varphi_\sigma(\boldsymbol{q}(t))\right\|^2, \quad t \in [t_{\sigma,i}, t_{\sigma,i+1}), \tag{5.69}$$

which is non-positive in σ. Since V_σ is positive-definite, Eq. 5.69 shows that V_σ strictly decreases and converges to the following set,

$$\mathcal{E} = \left\{\boldsymbol{q} \in \mathbb{R}^2 \mid \|\boldsymbol{\nabla}\varphi_\sigma(\boldsymbol{q})\| = \boldsymbol{0}\right\}, \tag{5.70}$$

which is the set of the equilibrium points of φ_σ. This completes the proof. □

Before we state the next theorem, which is the main result of this subsection, we define the set $\mathcal{L}$ as the union of all the stable manifolds l of all the non-zero norm critical points in $\mathcal{F}$.

Theorem 5.4.12. *Consider the switching closed-loop system in Eq. 5.34 with parameter k satisfying the condition in Eq. 5.67. Every trajectory of 5.34 that starts inside $\mathcal{F}$ and outside $\mathcal{L}$ asymptotically converges to the origin. This also implies that the origin is almost globally asymptotically stable.*

Proof. When a trajectory starts in $\mathcal{F}$ and outside $\mathcal{L}$, the functions V_σ create a sequence of strictly decreasing functions (Lemma 5.4.11) that have coincident values at the time instants that σ switches (Lemma 5.4.10). Taking into account Theorem 5.4.9 and Eq. 5.69, and invoking Theorem 3.1 in Liberzon (2003), the continuous sequence of strictly decreasing positive-definite functions shows asymptotic convergence of the trajectory to a critical point of $\varphi_\sigma, \sigma \in \{\mathcal{SS}, \mathcal{R}_0, \mathcal{R}_1, \cdots, \mathcal{R}_m\}$. Since the trajectory starts outside $\mathcal{L}$, this point cannot be a non-zero norm equilibrium point, and the origin is the only remaining candidate. This means that the trajectory asymptotically converges to the origin.

Also, the trajectories that start inside $\mathcal{L}$ converge to a non-zero norm equilibrium point. The elements of $\mathcal{L}$ are sets of measure zero in $\mathcal{F}$, which is the reason we cannot introduce a basin of attraction for non-zero norm equilibrium points. This implies that the origin is an almost globally asymptotically stable equilibrium point of the system in Eq. 5.34. □

5.4.4 Collision Avoidance Analysis

The robot's clearance from the obstacles and the domain's boundary is immediately concluded from Theorem 5.4.12. We state this conclusion in the following corollary.

Corollary 5.4.13. *Given the switching closed-loop system in Eq. 5.34 with parameter k satisfying the condition in Eq. 5.67, $\mathcal{F}$ is a positively invariant set for any trajectory that starts in $\mathcal{F}$, and consequently no collision occurs between the robot and the boundaries of the obstacles and the domain.*

Proof. We know $\varphi_\sigma \in [0,1) \, \forall \sigma \in \{\mathcal{SS}, \mathcal{R}_0, \mathcal{R}_1, \cdots, \mathcal{R}_m\}$, and consequently $\varphi_\sigma(\boldsymbol{q}(0)) \in [0,1)$ if $\boldsymbol{q}(0) \in \mathcal{F}$. The sequence of strictly decreasing positive functions V_σ in the proof of Theorem 5.4.12 shows that every φ_σ remains in the range $[0,1)$ for trajectories that start in $\mathcal{F}$ and outside $\mathcal{L}$ $\forall t \geq 0$. This implies that the trajectory stays in $\mathcal{F}$ for all future time, and the robot never hits the the obstacles' boundaries and the domain boundary, where $\varphi_{\mathcal{R}_i} = 1$. Also, trajectories that start in $\mathcal{L}$ monotonically converge to the corresponding non-zero norm equilibrium point due to the first-order dynamics of the robot. Thus, these trajectories never hit the obstacles' and the domain's boundaries either. This completes the proof. □

5.5 Simulation Results

In this section, we validate our theoretical results with MATLAB simulations of the motion of a holonomic robot with single-integrator dynamics. We consider

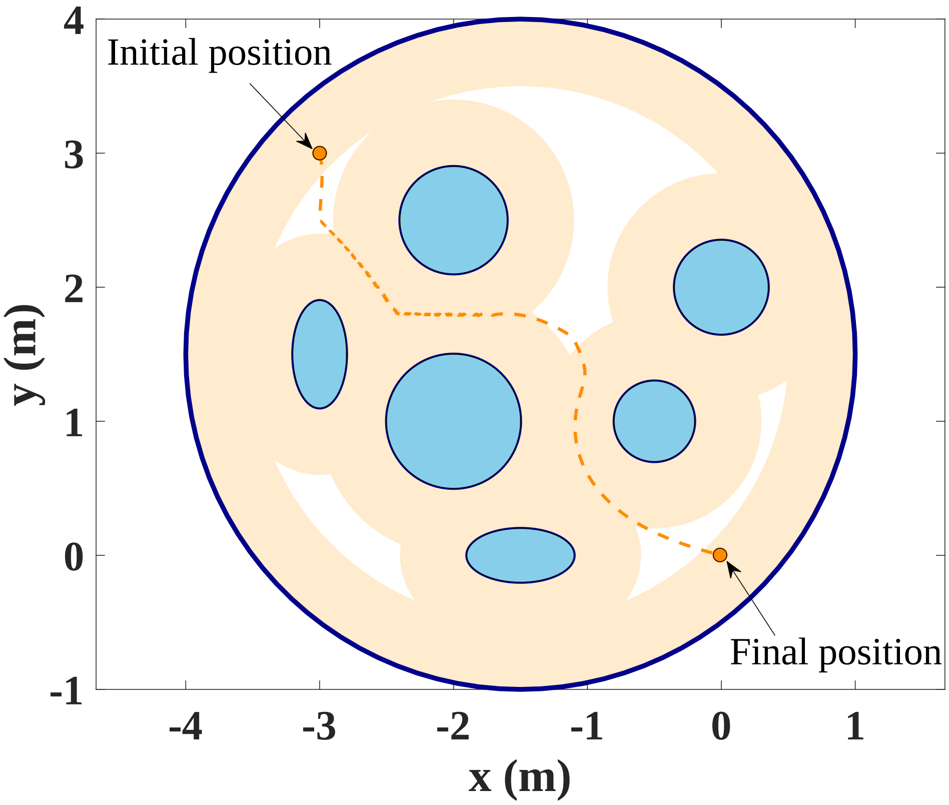


Figure 5.4: Trajectory of a Single Robot That Moves in a Bounded Domain When the Destination Is Located in a Repulsion Space.

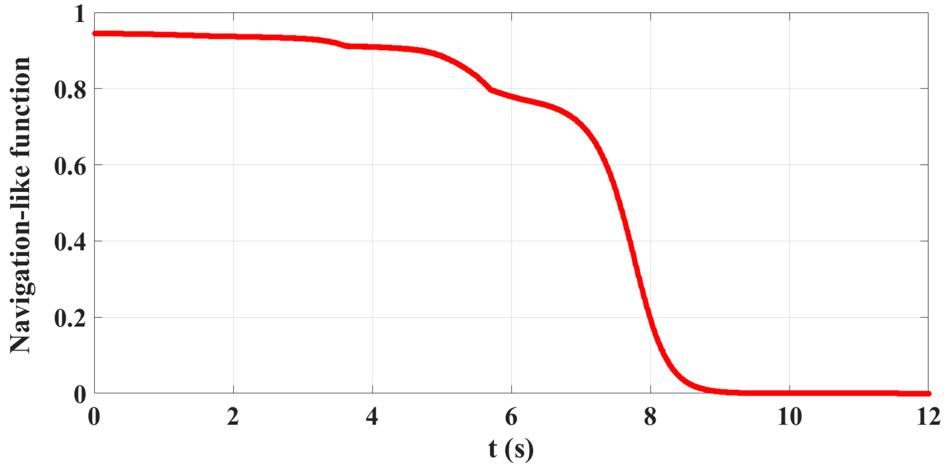


Figure 5.5: Time Evolution of the Navigation-Like Function for the Robot When the Destination Is Located in a Repulsion Space.

a circular domain that contains four circular obstacles and two elliptical obstacles, where the obstacles' configuration in the domain satisfies Assumption 5.1.9. The radius of the robot is $r = 10$ cm, and the size the domain is $r_{\mathcal{D}} = 2.5$ m. We set $k = 0.04$ and run two simulations.

In the first simulation, the robot's initial position is $\boldsymbol{q}(0) = [-3\ 3]^T$, and the destination is located in a repulsion space, which is $\mathcal{R}_0$ here. As shown in Fig. 5.4, the robot moves between the obstacles without colliding with them and converges to the destination (the origin) in almost 10 s. Also, the time evolution of the navigation-like function $\varphi_\sigma(\boldsymbol{q})$ is shown in Fig. 5.5. We can see that the NLF monotonically converges to zero and never becomes equal 1, which shows that the robot never hits the obstacles and the domain boundary.

In the second simulation, the robot starts from the same initial position, $\boldsymbol{q}(0) = [-3\ \ 3]^T$, while the domain and the obstacles are reconfigured in such a way that the origin is located in the safe space $\mathcal{SS}$. As shown in Fig. 5.6, the robot again converges to the destination without colliding with the obstacles. The time evolution of the navigation-like function $\varphi_\sigma(\boldsymbol{q})$ is also illustrated in Fig. 5.7 for this simulation. We again see that the value of the NLF monotonically converges to zero and never becomes equal to 1. This shows that the robot converges to the destination without colliding with the boundaries of the obstacles and the domain when the destination is located in the safe space.

We also see that the robot has a slow rate of convergence in both simulations, especially at the beginning of its motion. The reason is that the attraction force is much smaller than the repulsion force at the beginning. We know that the rate of convergence can be increased if we use smaller values for k, which would result in smaller repulsion forces. In this case, the robot's navigation through the environment would be "riskier," in that its clearance from the boundaries of the obstacles and the domain would decrease. Another way to modify the rate of convergence is to introduce an additional tuning parameter into the NLF's without affecting the types

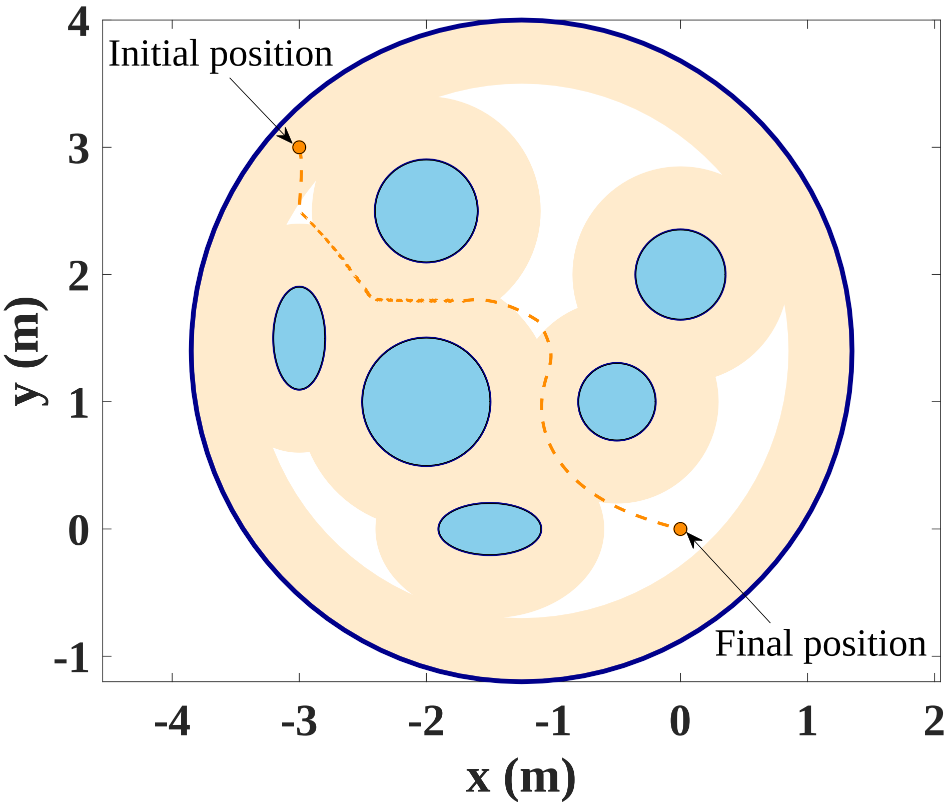


Figure 5.6: Trajectory of a Single Robot That Moves in a Bounded Domain When the Destination Is Located in the Safe Space.

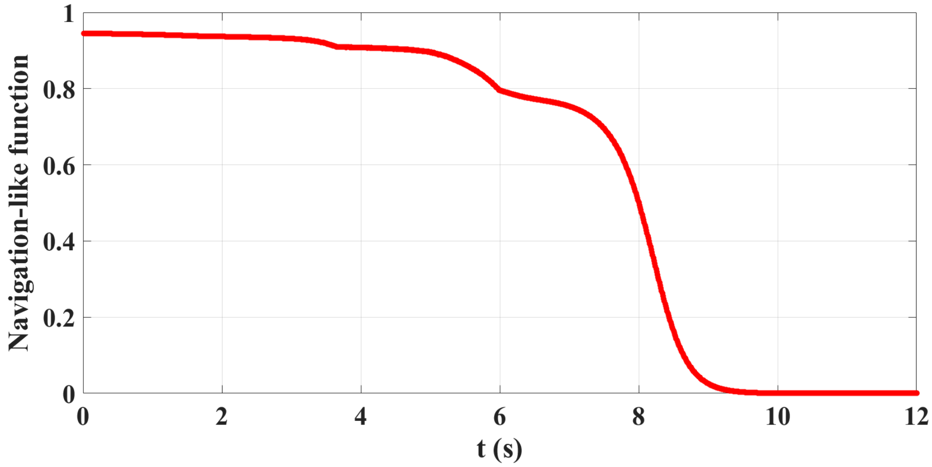


Figure 5.7: Time Evolution of the Navigation-Like Function for the Robot When the Destination Is Located in the Safe Space.

of the critical points and their stability characteristics. This is a subject of ongoing work.

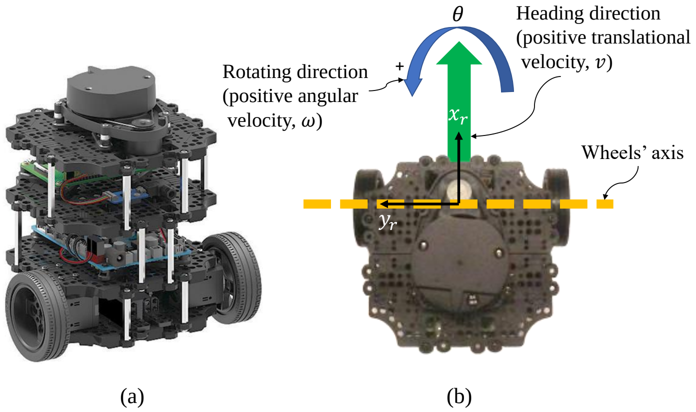


Figure 5.8: (a) 3-D View of the Turtlebot 3 Burger Robot, and (b) Overhead View with Body-Fixed Coordinate Frame.

5.6 Experimental Implementation and Results

In this section, we present experimental results of implementing our proposed obstacle avoidance controller on a physical robot in order to evaluate performance of the controller in practice. We use a commercial nonholonomic robot called the Turtlebot3 Burger robot, which is shown in Fig. 5.8, for the experimental tests. Since the controller is designed for a holonomic robot, we need to adapt the proposed controller in such a way that it can be applied to a nonholonomic robot. To this end, we use the adaptation method in (Lafmejani *et al.*, 2020). We consider two scenarios. In each scenario, the environment contains convex obstacles, whose shapes are arbitrary and unknown to the robot. The robot must navigate between the obstacles without any collisions with them and converge to the goal position (the origin).

5.6.1 Scenario 1

In the first scenario, we demonstrate stabilization of the robot to the goal point by our proposed obstacle avoidance controller in the presence of convex obstacles in the environment. We set the controller gain to $k = 0.15$. Figure 5.9 shows snapshots of

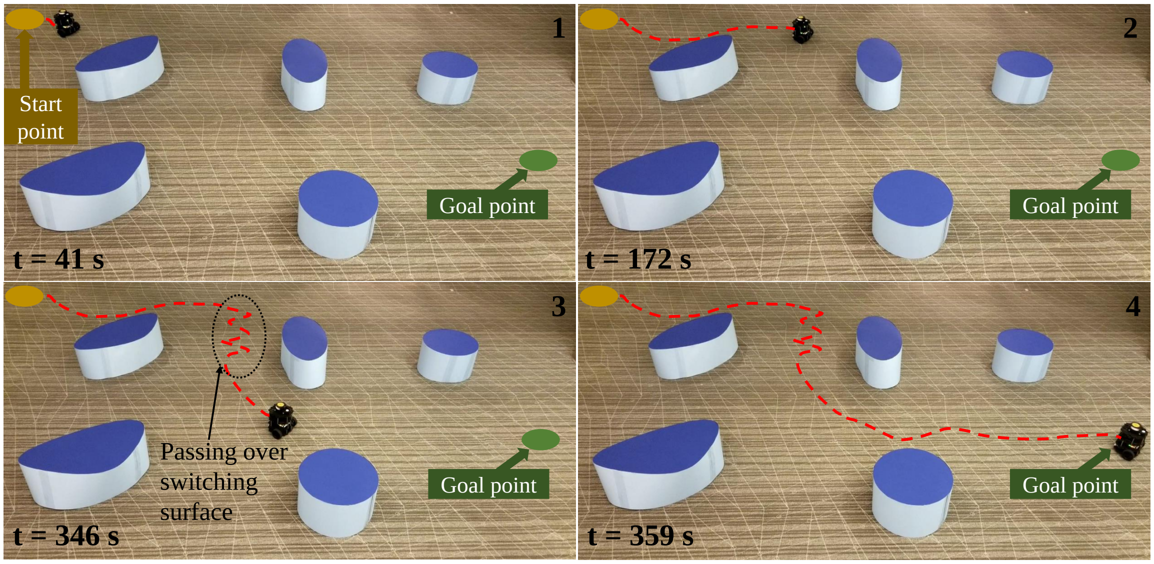


Figure 5.9: Snapshots of the Experimental Implementation of the First Scenario. The Controller Gains Are Set to $k = 0.15$.

the experimental implementation of the first scenario. The Burger robot starts at the initial point $\boldsymbol{q}(0) = [-3.75 \quad 2.0]^{\mathrm{T}}$ m. The goal point is the origin of the global frame. As shown in the third snapshot, the robot has to pass over the switching surface in order to navigate between the two closest obstacles without colliding with them.

5.6.2 Scenario 2

In the second scenario, we illustrate the effect of the parameter k on the performance of the robot on stabilizing to the goal point. We also change the obstacles' configuration. Two different cases are tested with this configuration of obstacles in the environment but with different values of the parameter k. The robot's initial position is $\boldsymbol{q}(0) = [-3.75 \quad 2.25]^{\mathrm{T}}$ m in both cases. We first set $k = 0.12$. Figure 5.10 shows snapshots of the experimental results of the second scenario when $k = 0.12$. We see that the robot converges to the destination in 198 s. We then set $k = 0.1$. The result is shown in Fig. 5.11, where we see that the robot converges to the destination in 124 s. This demonstrates that the robot stabilizes to the goal point faster when

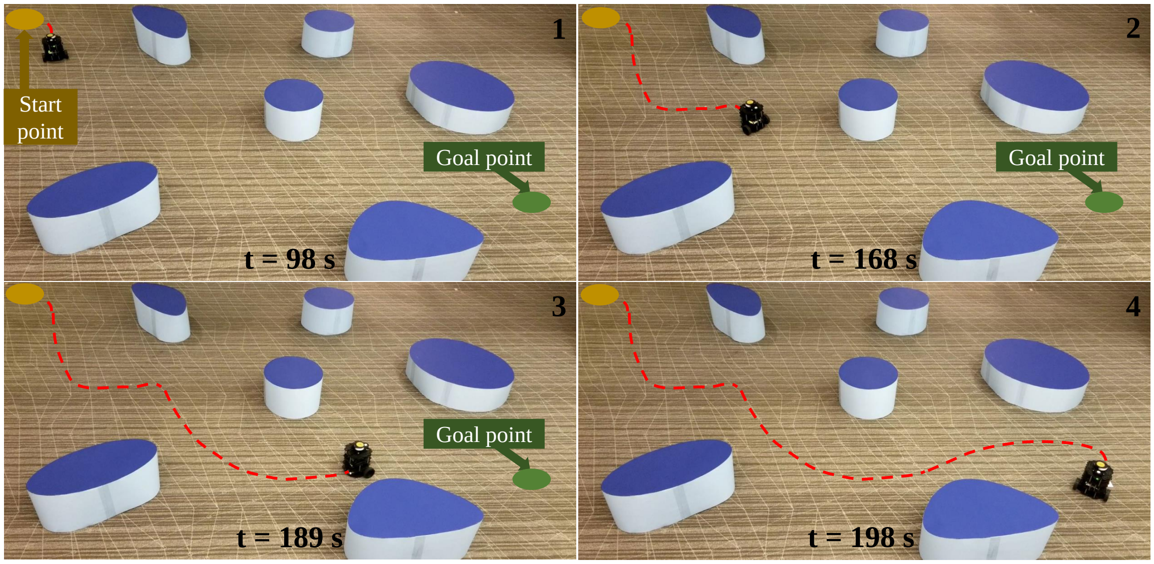


Figure 5.10: Snapshots of the Experimental Implementation of the Second Scenario. The Controller Gains Are Set to $k = 0.12$.

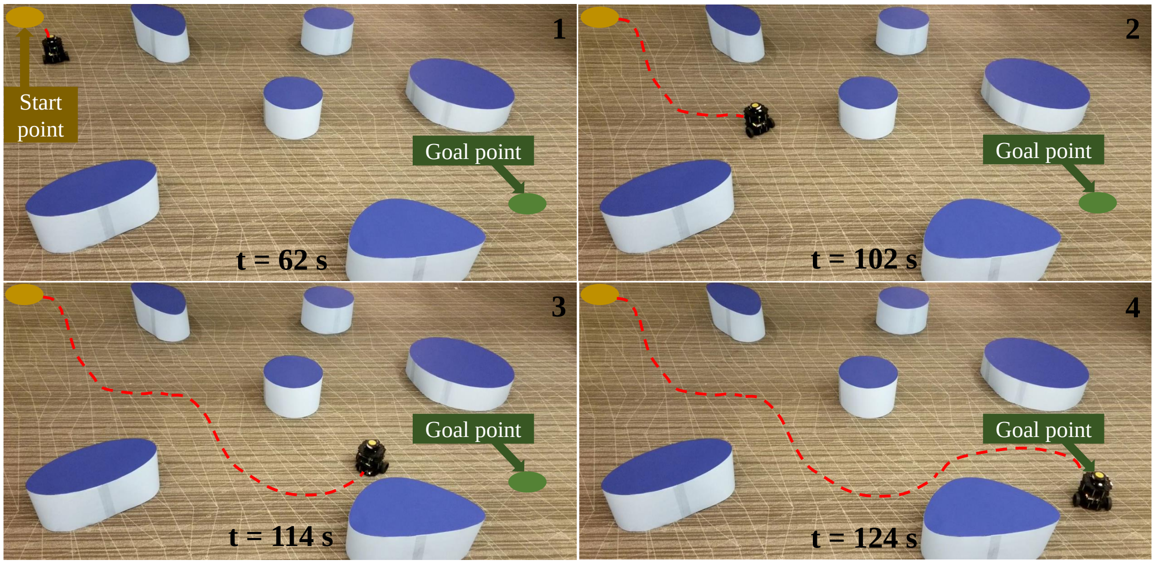


Figure 5.11: Snapshots of the Experimental Implementation of the Second Scenario. The Controller Gains Are Set to $k = 0.1$.

we use a smaller value of the parameter k.

5.7 Application to Position Control of Collective Transport in Bounded Domains with Convex Obstacles

In this section, we propose a decentralized controller for collective transport by a team of robots that have double-integrator dynamics and move in a bounded convex domain with convex obstacles. To design the controller, we use the obstacle avoidance controller that we designed for a single robot in a bounded domain in the preceding section.

5.7.1 Problem Statement

We consider a team of N identical point-mass robots that move in a bounded planar convex domain and are rigidly attached to a payload in an arbitrary configuration, as shown in Fig. 5.12. We assume that each robot has access to its own position and velocity with respect to an inertial coordinate system, which is common to all the robots. The robots do not communicate with one another and are not assigned predefined trajectories. They also lack information about the payload's kinematics and dynamics, the number of robots in the transport team, and the robots' distribution around the payload. We also assume that the domain contains convex obstacles, and that the positions and shapes of the obstacles are unknown to the robots. We make the following assumptions about the configuration of the obstacles in the domain and the robots' distribution around the payload.

Assumption 5.7.1. *The space between each pair of obstacles and the space between an obstacle and the boundary of the domain is larger than the largest dimension of the payload.*

Assumption 5.7.2. *The payload is contained in the convex polygon that has the robots as the vertices.*

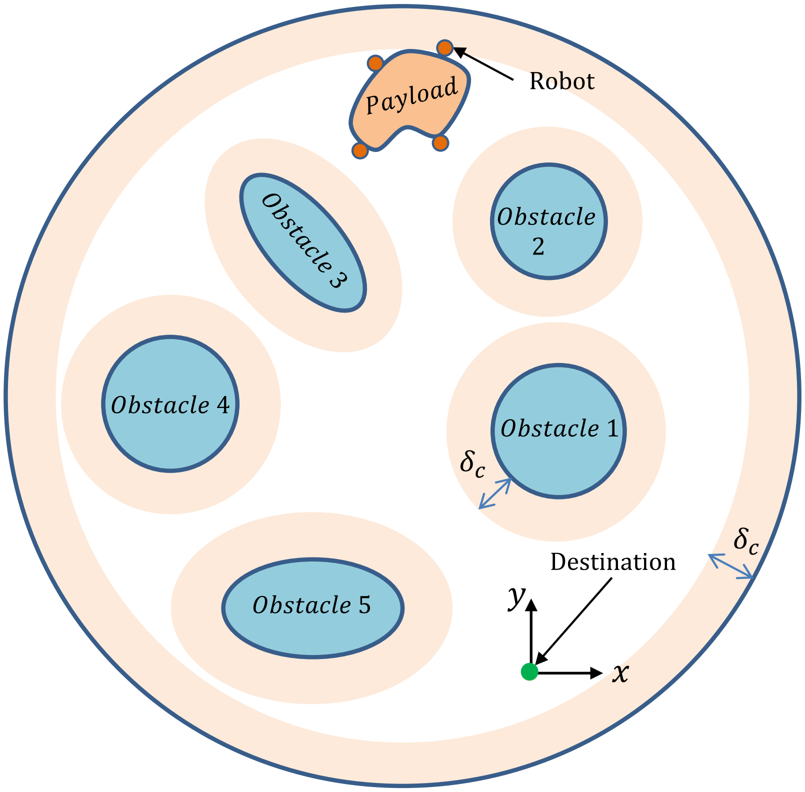


Figure 5.12: A Schematic Representation of a Collective Transport Task with Four Robots in a Bounded Convex Domain.

Assumption 5.7.1 allows the payload to be physically transportable through any space in the domain. Also, Assumption 5.7.2 guarantees that the payload never collides with the obstacles if the robots never collide with the obstacles.

We define $\boldsymbol{x}_o = [x_o \ y_o]^T \in \mathbb{R}^2$ and $\theta_o \in \mathbb{R}$ as the position of the payload's center of mass, point O in Fig. 2.11, and the payload's orientation with respect to a global coordinate frame, respectively. We define $\boldsymbol{x}_i = [x_i \ y_i]^T \in \mathbb{R}^2$ as the position of robot i and $\boldsymbol{x}_d = [x_d \ y_d]^T \in \mathbb{R}^2$ as the position of the target point in the global frame, as shown in Fig. 2.12. The center of mass of the entire system, including both the load and the robots, is denoted by point C in Fig. 2.11. Given that points O and C are not necessarily coincident, we define $\boldsymbol{x}_c = [x_c \ y_c]^T \in \mathbb{R}^2$ as the position of C in the global frame and $\mathbf{r}_c \in \mathbb{R}^2$ as the vector from C to O, as shown in Fig. 2.11. We also define $\boldsymbol{r}_i = [r_{ix} \ r_{iy}]^T \in \mathbb{R}^2$ as the vector from C to the attachment point of robot i in the global frame.

Each robot i knows its own position $\boldsymbol{x}_i$ and velocity $\dot{\boldsymbol{x}}_i$ and applies an actuating force $\boldsymbol{u}_i = [u_{ix} \; u_{iy}]^T \in \mathbb{R}^2$ to the payload. The control objective is to design the forces $\boldsymbol{u}_i$, $i = 1, ..., N$, such that the robots drive the position of the payload's center of mass, $\boldsymbol{x}_o$, to the target position $\boldsymbol{x}_d$ while colliding collisions with the obstacles. The only sensor feedback available to the robots consists of their on-board measurements of their own positions and velocities and their distance from the obstacles in their sensing range.

5.7.2 Dynamical Model

To derive the dynamical model of the entire system, comprised of both the load and the robots, we use the framework in Chapter 4. We denote the mass of each robot and the mass of the payload by m_r and m_o, respectively. We also define I_o as the payload's moment of inertia about the axis perpendicular to the plane and passing through O. Considering the entire system as a rigid body and defining $\boldsymbol{q} := [x_c \; y_c \; \theta_o]^T \in \mathbb{R}^3$ as the vector of generalized coordinates, we can write the equation of motion of the entire system as

$$\begin{bmatrix} m\boldsymbol{I} & 0 \\ 0 & I \end{bmatrix} \ddot{\boldsymbol{q}} = \begin{bmatrix} \boldsymbol{I} & \cdots & \boldsymbol{I} \\ \hat{\boldsymbol{r}}_1^T & \cdots & \hat{\boldsymbol{r}}_N^T \end{bmatrix} \boldsymbol{u}, \tag{5.71}$$

where m and I are the mass and moment of inertia of the entire system, given by:

$$\begin{aligned} m &= m_o + N m_r, \\ I &= I_o + m_o \left\|\mathbf{r}_c\right\|^2 + m_r \sum_{i=1}^{N} \left\|\mathbf{r}_i\right\|^2, \end{aligned} \tag{5.72}$$

and $\hat{\boldsymbol{r}}_i \in \mathbb{R}^2$ and $\boldsymbol{u} \in \mathbb{R}^{2N}$ are defined as

$$\hat{\boldsymbol{r}}_i = [-r_{iy} \;\; r_{ix}]^T, \tag{5.73}$$

$$\boldsymbol{u} = [\boldsymbol{u}_1^T \; \cdots \; \boldsymbol{u}_N^T]^T. \tag{5.74}$$

The matrix $\boldsymbol{I} \in \mathbb{R}^{2\times 2}$ is the identity matrix.

5.7.3 Controller Design

In this section, we present decentralized robot controllers for the system described by Eq. 3.52 that produce convergence of the payload's center of mass to a neighborhood of the desired position $\boldsymbol{x}_d$ while guaranteeing its clearance from the obstacles and the domain's boundary. The proposed control law is written as

$$\boldsymbol{u}_i = -\boldsymbol{K}_d \dot{\boldsymbol{x}}_i - \boldsymbol{K}_p \boldsymbol{\nabla}\varphi(\boldsymbol{x}_i), \tag{5.75}$$

in which $\boldsymbol{K}_p = K_p \boldsymbol{I}$ and $\boldsymbol{K}_d = K_d \boldsymbol{I}$ are gain matrices, where K_p and K_d are strictly positive constants. Also, the term $\boldsymbol{\nabla}\varphi(\boldsymbol{x}_i)$ is the gradient of the NLF at the position of robot i. This control law implies that each robot *selfishly* tries to stabilize its own position to the target position using the NLF that is calculated at its own position. Since the robots are attached to distinct points on the payload's boundary, convergence of all the robots' positions to the target position is impossible. However, by each applying the decentralized controller in Eq. 5.75, the robots produce a collective transport behavior that approximately achieves the control objective defined in Section 5.7.1. We analyze and discuss this behavior in the next section.

5.7.4 Motion Analysis

To analyze the collective behavior of the entire system of the payload and the robots with the proposed controller, we first derive the dynamics of the closed-loop system and then investigate the stability and convergence properties of this system.

Closed-loop dynamics

There is a holonomic kinematic constraint between the position of robot i and the position of the system's center of mass (see Fig. 2.12), given by

$$\boldsymbol{x}_i = \boldsymbol{x}_c + \boldsymbol{r}_i. \tag{5.76}$$

Taking the time derivative of Eq. 5.76, we obtain

$$\dot{\boldsymbol{x}}_i = \dot{\boldsymbol{x}}_c + \hat{\boldsymbol{r}}_i \dot{\theta}_o, \tag{5.77}$$

where $\hat{\boldsymbol{r}}_i$ is given by Eq. 5.73. Substituting the expressions for $\boldsymbol{x}_i$ and $\dot{\boldsymbol{x}}_i$ in Eqs. 5.76 and 5.77 into Eq. 5.75, we obtain

$$\boldsymbol{u}_i = -\boldsymbol{K}_d(\dot{\boldsymbol{x}}_c + \hat{\boldsymbol{r}}_i \dot{\theta}_o) - \boldsymbol{K}_p \boldsymbol{\nabla}\varphi(\boldsymbol{x}_i). \tag{5.78}$$

We now incorporate the decentralized control law for $\boldsymbol{u}_i$ in Eq. 5.78 into the dynamical model in Eq. 5.71 to derive the equation of motion of the closed-loop system as

$$\begin{aligned} \boldsymbol{M}\ddot{\boldsymbol{x}}_c &= -\boldsymbol{K}_d \sum_{i=1}^{N} (\dot{\boldsymbol{x}}_c + \hat{\boldsymbol{r}}_i \dot{\theta}_o) - \boldsymbol{K}_p \sum_{i=1}^{N} \boldsymbol{\nabla}\varphi(\boldsymbol{x}_i), \\ I\ddot{\theta}_o &= -K_d \sum_{i=1}^{N} \hat{\boldsymbol{r}}_i^T (\dot{\boldsymbol{x}}_c + \hat{\boldsymbol{r}}_i \dot{\theta}_o) - K_p \sum_{i=1}^{N} \hat{\boldsymbol{r}}_i^T \boldsymbol{\nabla}\varphi(\boldsymbol{x}_i), \end{aligned} \tag{5.79}$$

where $\boldsymbol{M} = m\boldsymbol{I}$. Taking into account the facts that $\boldsymbol{r}_i \times \boldsymbol{r}_i = \boldsymbol{0}$ and $[\boldsymbol{r}_i^T \ 0]^T \times [\boldsymbol{a}^T \ 0]^T = [0 \ \ 0 \ \ \hat{\boldsymbol{r}}_i^T \boldsymbol{a}]^T$, where $\boldsymbol{a}$ is an arbitrary vector in $\mathbb{R}^2$, the closed-loop system in Eq. 5.79 can be rewritten as

$$\begin{aligned} \boldsymbol{M}\ddot{\boldsymbol{x}}_c &= -N\boldsymbol{K}_d \dot{\boldsymbol{x}}_c - \boldsymbol{K}_d \sum_{i=1}^{N} \hat{\boldsymbol{r}}_i \dot{\theta}_o - \boldsymbol{K}_p \sum_{i=1}^{N} \boldsymbol{\nabla}\varphi(\boldsymbol{x}_i), \\ I\ddot{\theta}_o &= -K_d \sum_{i=1}^{N} \hat{\boldsymbol{r}}_i^T \dot{\boldsymbol{x}}_c - K_d \rho \dot{\theta}_o - K_p \sum_{i=1}^{N} \hat{\boldsymbol{r}}_i^T \boldsymbol{\nabla}\varphi(\boldsymbol{x}_i), \end{aligned} \tag{5.80}$$

where $\rho := \sum_{i=1}^{N} \|\boldsymbol{r}_i\|^2$. For notational simplicity, we also define $\boldsymbol{\varrho} := \sum_{i=1}^{N} \boldsymbol{r}_i$, which implies that $\hat{\boldsymbol{\varrho}} := \sum_{i=1}^{N} \hat{\boldsymbol{r}}_i$.

Note that $\boldsymbol{r}_i$ changes with the payload's rotation, and we can write $\boldsymbol{r}_i = \boldsymbol{r}_i(\theta_o)$. The magnitude of $\boldsymbol{r}_i$ however remains unchanged since each robot i is rigidly attached to the payload and C is a fixed point on the payload. Consequently, when the payload rotates, the directions of $\boldsymbol{\varrho}$ and $\hat{\boldsymbol{\varrho}}$ change while their magnitudes remain constant.

Equilibrium points

The equilibrium state of the closed-loop system in Eq. 5.80 is obtained by setting $\ddot{\boldsymbol{x}}_c = \dot{\boldsymbol{x}}_c = \mathbf{0}$ and $\ddot{\theta}_o = \dot{\theta}_o = 0$, which by using Eq. 5.76, results in the following equations:

$$\sum_{i=1}^{N} \boldsymbol{\nabla}\varphi\left(\boldsymbol{x}_c^* + \boldsymbol{r}_i(\theta_o^*)\right) = \mathbf{0}, \tag{5.81}$$

$$\sum_{i=1}^{N} \hat{\boldsymbol{r}}_i^T(\theta_o^*)\boldsymbol{\nabla}\varphi\left(\boldsymbol{x}_c^* + \boldsymbol{r}_i(\theta_o^*)\right) = 0, \tag{5.82}$$

in which the superscript * denotes the equilibrium state. We must solve Eq. 5.81 and Eq. 5.82 simultaneously to find the values of $\boldsymbol{x}_c$ and θ_o at the equilibrium points. Regarding the equation of the navigation function and its gradient, these equations are very complicated to solve analytically. Hence, we use Eq. 5.81 and Eq. 5.82 as the representation of the equilibrium points of the system in Eq. 5.80.

Stability characteristics of the equilibrium points

To analyze the stability characteristics of the equilibrium points of the closed-loop system in Eq. 5.80, we use Lyapunov's indirect method and linearize Eq. 5.80 around the equilibrium points represented by Eq. 5.81 and Eq. 5.82. Taking into account Eq. 5.76 and the fact that $\boldsymbol{r}_i = \boldsymbol{r}_i(\theta_o)$, we can define $\boldsymbol{X} = [\boldsymbol{X}_1^T \ \boldsymbol{X}_2^T]^T \in \mathbb{R}^6$ as the state vector for the system in Eq. 5.80, where $\boldsymbol{X}_1 := [\boldsymbol{x}_c^T \ \theta_o]^T \in \mathbb{R}^3$ and $\boldsymbol{X}_2 := [\dot{\boldsymbol{x}}_c^T \ \dot{\theta}_o]^T \in \mathbb{R}^3$. Given $\boldsymbol{X}$, the state-space representation of Eq. 5.80 is written as

$$\dot{\boldsymbol{X}} = \boldsymbol{F}(\boldsymbol{X}) := \begin{bmatrix} \boldsymbol{X}_2 \\ -\boldsymbol{H}^{-\mathbf{1}}\left(\boldsymbol{G}_1(\theta_o)\boldsymbol{X}_2 + \boldsymbol{G}_2(\boldsymbol{X}_1)\right) \end{bmatrix}, \tag{5.83}$$

where $\boldsymbol{H} = diag(m, m, I) \in \mathbb{R}^{3\times3}$, and $\boldsymbol{G}_1 \in \mathbb{R}^{3\times3}$ and $\boldsymbol{G}_2 \in \mathbb{R}^3$ are given by

$$\boldsymbol{G}_1(\theta_o) = K_d \begin{bmatrix} N\boldsymbol{I} & \hat{\boldsymbol{\varrho}} \\ \hat{\boldsymbol{\varrho}}^T & \rho \end{bmatrix}, \tag{5.84}$$

$$\boldsymbol{G}_2 = K_p \sum_{i=1}^{N} \left(\begin{bmatrix} \boldsymbol{\nabla}\varphi(\boldsymbol{x}_i) \\ \hat{\boldsymbol{r}}_i^T(\theta_o)\boldsymbol{\nabla}\varphi(\boldsymbol{x}_i) \end{bmatrix} \right). \tag{5.85}$$

To derive the linearized model, we must calculate the derivative of $\boldsymbol{F}(\boldsymbol{X})$ with respect to $\boldsymbol{X}$ as

$$\frac{\partial \boldsymbol{F}(\boldsymbol{X})}{\partial \boldsymbol{X}} = \begin{bmatrix} \mathbf{0}_{3\times3} & \boldsymbol{I}_{3\times3} \\ -\boldsymbol{H}^{-\mathbf{1}}\left(\frac{\partial \boldsymbol{G}_1(\theta_o)}{\partial \boldsymbol{X}_1}\boldsymbol{X}_2 + \frac{\partial \boldsymbol{G}_2(\boldsymbol{X}_1)}{\partial \boldsymbol{X}_1}\right) & -\boldsymbol{H}^{-\mathbf{1}}\boldsymbol{G}_1 \end{bmatrix}. \tag{5.86}$$

The linearized model is obtained by evaluating the right-hand side of Eq. 5.86 at the equilibrium points, where $\boldsymbol{X}_1 = \boldsymbol{X}_1^*$ and $\boldsymbol{X}_2 = \mathbf{0}$. Hence, the linearized form of Eq. 5.83 is obtained as

$$\dot{\boldsymbol{X}} = \begin{bmatrix} \mathbf{0}_{3\times3} & \boldsymbol{I}_{3\times3} \\ -\boldsymbol{H}^{-\mathbf{1}}\left(\frac{\partial \boldsymbol{G}_2(\boldsymbol{X}_1)}{\partial \boldsymbol{X}_1}\right) & -\boldsymbol{H}^{-\mathbf{1}}\boldsymbol{G}_1 \end{bmatrix}_{\boldsymbol{X}_1 = \boldsymbol{X}_1^*} \boldsymbol{X}, \tag{5.87}$$

where $\boldsymbol{X}_1^* = [\boldsymbol{x}_c^{*^T} \theta_o^*]^T$ described by Eq. 5.81 and Eq. 5.82. To study the stability properties of the equilibrium points, we must analyze the signs of the eigenvalues of the matrix in the right-hand side of Eq. 5.87. We know that $\boldsymbol{H}$ is positive definite and can confirm that $\boldsymbol{G}_1$ is positive definite too. By *Lemma 3.5* in (Koditschek, 1989), we know that the linear system in Eq. 5.87 has the stability properties of the linear system defined by

$$\dot{\boldsymbol{X}}_1 = -\boldsymbol{H}^{-1}\boldsymbol{A}\boldsymbol{X}_1, \tag{5.88}$$

where

$$\boldsymbol{A} := \left(\frac{\partial \boldsymbol{G}_2(\boldsymbol{X}_1)}{\partial \boldsymbol{X}_1} \right)_{\boldsymbol{X}_1 = \boldsymbol{X}_1^*} \tag{5.89}$$

Toward this end, we define $\boldsymbol{A}_i(\boldsymbol{X}_1) \in \mathbb{R}^{3\times 3}$ as

$$\boldsymbol{A}_i(\boldsymbol{X}_1) := \frac{\partial}{\partial \boldsymbol{X}_1} \left(\begin{bmatrix} \boldsymbol{\nabla}\varphi(\boldsymbol{x}_i) \\ \hat{\boldsymbol{r}}_i^T(\theta_o)\boldsymbol{\nabla}\varphi(\boldsymbol{x}_i) \end{bmatrix} \right). \tag{5.90}$$

Taking into account Eq. 5.85, we can confirm that

$$\boldsymbol{A} = K_p \sum_{i=1}^{N} \boldsymbol{A}_i(\boldsymbol{X}_1 = \boldsymbol{X}_1^*). \tag{5.91}$$

By the chain rule, we can rewrite $\boldsymbol{A}_i$ in Eq. 5.90 as

$$\boldsymbol{A}_i(\boldsymbol{X}_1) = \begin{bmatrix} \frac{\partial}{\partial \boldsymbol{x}_i}\left(\boldsymbol{\nabla}\varphi(\boldsymbol{x}_i)\right)\frac{\partial \boldsymbol{x}_i}{\partial \boldsymbol{X}_1} \\ \hat{\boldsymbol{r}}_i^T \frac{\partial}{\partial \boldsymbol{x}_i}\left(\boldsymbol{\nabla}\varphi(\boldsymbol{x}_i)\right)\frac{\partial \boldsymbol{x}_i}{\partial \boldsymbol{X}_1} + \boldsymbol{\nabla}\varphi(\boldsymbol{x}_i)^T \frac{\partial \hat{\boldsymbol{r}}_i}{\partial \boldsymbol{X}_1} \end{bmatrix}. \tag{5.92}$$

We know that $\frac{\partial}{\partial \boldsymbol{x}_i}(\boldsymbol{\nabla}\varphi(\boldsymbol{x}_i)) = \boldsymbol{\nabla}^2\varphi(\boldsymbol{x}_i)$ and can also confirm from Eq. 5.76 that

$$\frac{\partial \boldsymbol{x}_i}{\partial \boldsymbol{X}_1} = \begin{bmatrix} \boldsymbol{I}_{2\times 2} & \hat{\boldsymbol{r}}_i \end{bmatrix} \tag{5.93}$$

$$\frac{\partial \hat{\boldsymbol{r}}_i}{\partial \boldsymbol{X}_1} = \begin{bmatrix} \boldsymbol{0}_{2\times 2} & -\boldsymbol{r}_i \end{bmatrix}. \tag{5.94}$$

Incorporating Eq. 5.93 and Eq. 5.94 into Eq. 5.92, we obtain

$$\boldsymbol{A}_i = \begin{bmatrix} \boldsymbol{\nabla}^2\varphi(\boldsymbol{x}_i) & \boldsymbol{\nabla}^2\varphi(\boldsymbol{x}_i)\hat{\boldsymbol{r}}_i \\ \hat{\boldsymbol{r}}_i^T\boldsymbol{\nabla}^2\varphi(\boldsymbol{x}_i) & \hat{\boldsymbol{r}}_i^T\boldsymbol{\nabla}^2\varphi(\boldsymbol{x}_i)\hat{\boldsymbol{r}}_i - \hat{\boldsymbol{r}}_i^T\boldsymbol{\nabla}\varphi(\boldsymbol{x}_i) \end{bmatrix}. \tag{5.95}$$

Using Eq. 5.91 and Eq. 5.82, we calculate the matrix $\boldsymbol{A}$ as

$$\boldsymbol{A} = K_p \sum_{i=1}^{N} \begin{bmatrix} \boldsymbol{\nabla}^2\varphi(\boldsymbol{x}_i) & \boldsymbol{\nabla}^2\varphi(\boldsymbol{x}_i)\hat{\boldsymbol{r}}_i \\ \hat{\boldsymbol{r}}_i^T\boldsymbol{\nabla}^2\varphi(\boldsymbol{x}_i) & \hat{\boldsymbol{r}}_i^T\boldsymbol{\nabla}^2\varphi(\boldsymbol{x}_i)\hat{\boldsymbol{r}}_i \end{bmatrix}. \tag{5.96}$$

To determine the signs of the eigenvalues of $\boldsymbol{A}$, we need to have knowledge about the signs of the eigenvalues of each matrix $\boldsymbol{A}_i$ at the equilibrium state, which is very complicated because of the arbitrary geometry of the payload and also because of the highly nonlinear structure of the $\varphi(\boldsymbol{x}_i)$ functions. Hence, we leave the rest of this analysis for our future work.

5.7.5 Simulation Results

We validate our theoretical results with MATLAB simulations of collective transport of a circular payload by four point-mass robots. We consider a circular domain that contains four circular obstacles, where the obstacles' configuration in the domain satisfies Assumption 5.7.1 and Assumption 5.7.2.

In the first simulation, the robots have a uniform distribution around the payload and must transport it to the origin while avoiding the obstacles. The initial position of the payload's center of mass is $\boldsymbol{x}_o(0) = [-1\ 3]^T$. We set $K_p = 1$ and $K_d = 0.6$. As shown in Fig. 5.13, the robots steer the payload between the obstacles without colliding with them and drive the payload's center of mass to the destination (the origin that is shown in green). Also, the time evolution of the navigation-like functions that are associated with each robot are shown in Fig. 5.14. We can see that each NLF has almost monotonic decrease with time and never becomes equal to 1, which shows that none of the robots contact an obstacle. Figure 5.14 also shows that the payload's center of mass converges to the destination in almost 16 seconds. In the second simulation, the robots have a nonuniform distribution around the payload and must transport it to the origin while avoiding the obstacles. The initial position of the payload's center of mass is changed to $\boldsymbol{x}_o(0) = [-3\ \ 2.25]^T$, and the controller gains are the same as those in the first simulation. As shown in Fig. 5.15, the robots again transport the payload toward the destination and avoid collisions with the obstacles. The time evolution of the navigation-like functions in Fig. 5.16 show that each robot almost monotonically converges to a vicinity of the origin while avoiding collisions with the obstacles. We also see that the motion becomes very slow between $t = 10\ s$ and $t = 14\ s$. The reason is that the robots detect three obstacles in their sensing radius when they are in the area in the middle of the domain. The robots apply

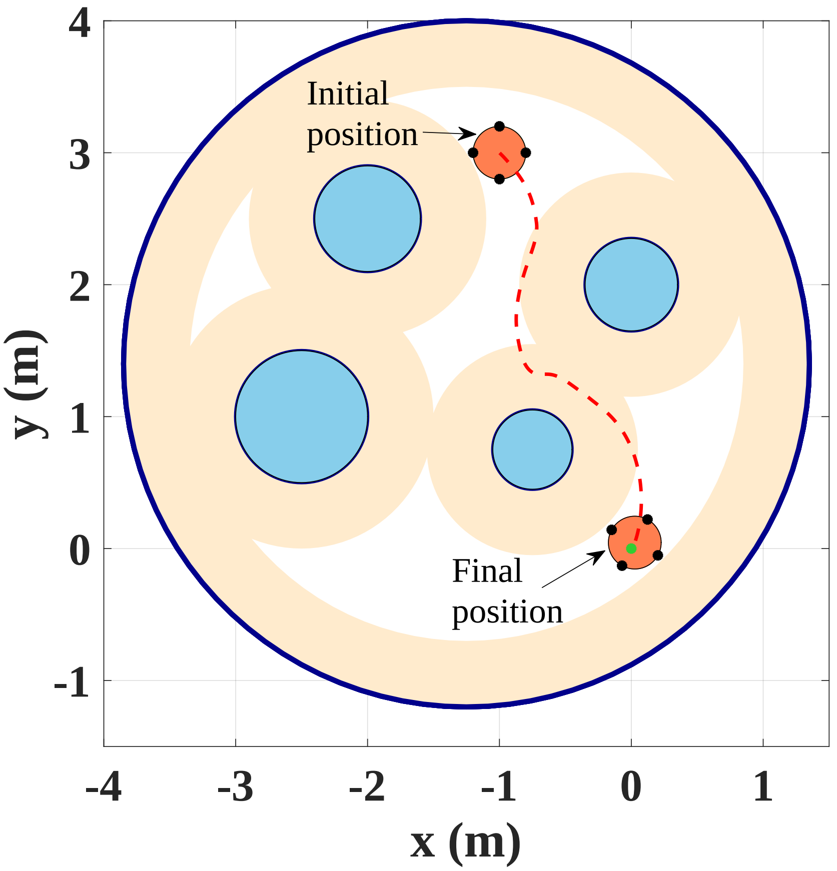


Figure 5.13: Snapshots of a Collective Transport Task in a Bounded Domain by Four Robots with a Uniform Distribution Around the Payload.

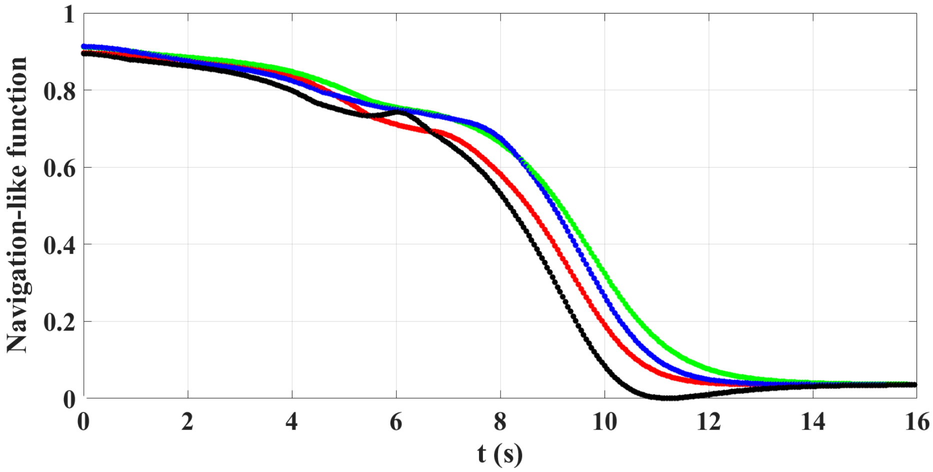


Figure 5.14: Time Evolution of the Navigation-Like Functions for Four Robots with a Uniform Distribution Around the Payload.

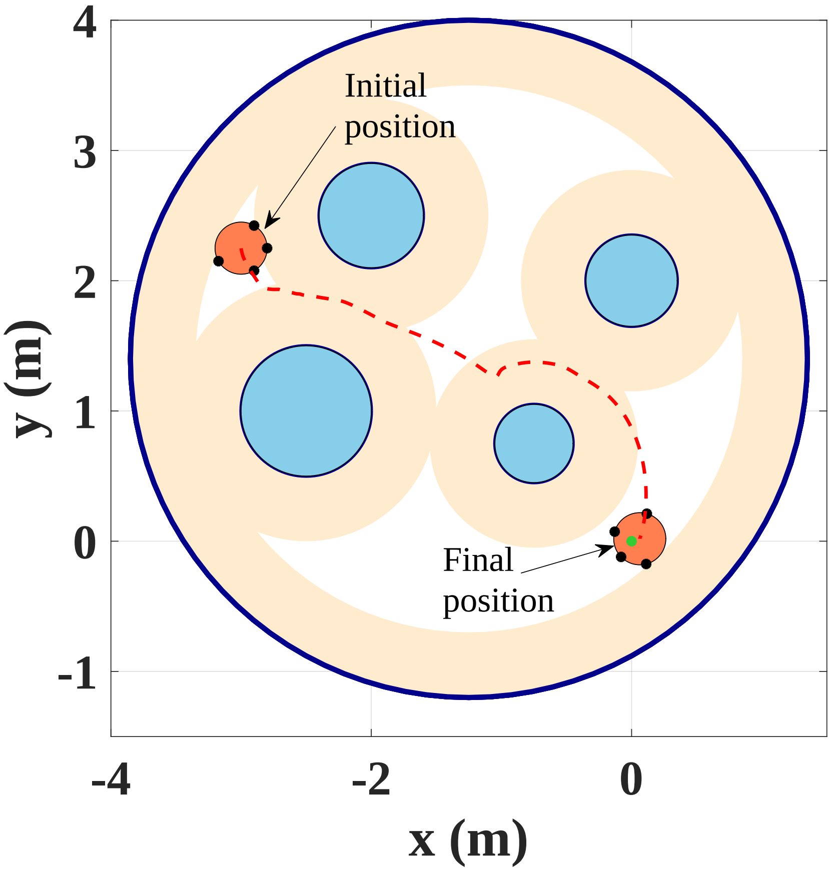


Figure 5.15: Snapshots of a Collective Transport Task in a Bounded Domain by Four Robots with a Nonuniform Distribution Around the Payload.

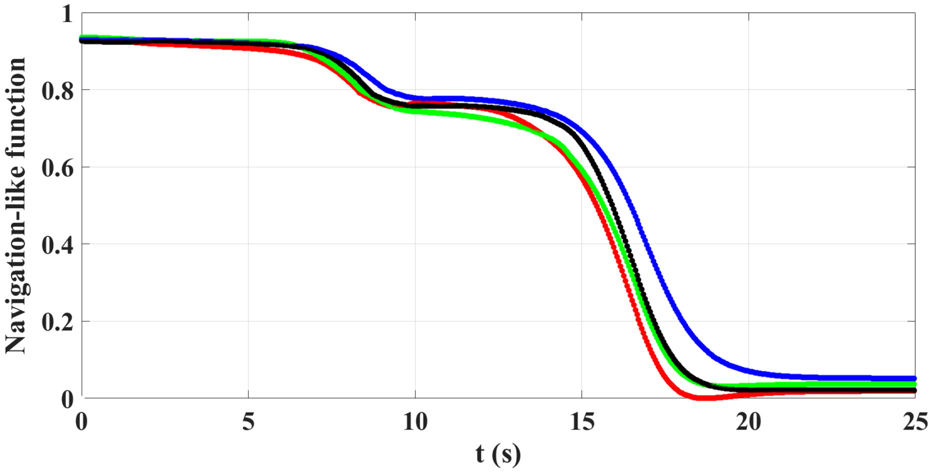


Figure 5.16: Time Evolution of the Navigation-Like Functions for Four Robots with a Nonuniform Distribution Around the Payload.

opposing repulsion forces to the payload, which slow down its motion. Since there is no local minimum point in that area, the robots do not become stuck, and instead circumvent the obstacles on their way toward the destination. Figure 5.16 also shows that the payload's center of mass converges to a neighborhood of the origin in almost 21 seconds.

Chapter 6

CONCLUSION AND FUTURE WORK

In this chapter, we conclude this thesis and mention some possible directions for future work.

In Chapter 2, we first presented a decentralized control strategy for multi-robot collective transport based on sliding mode control. The controllers did not require inter-robot communication, knowledge of the load dynamics and geometry, or the size and spatial configuration of the transport team. We validated this control strategy in simulations with point-mass robots and with 3D models of robots with realistic dynamics, as well as in experiments with a team of small mobile robots. The simulations and experiments demonstrated the effectiveness of the strategy at driving the transport team to a target speed in a desired direction. We then presented decentralized proportional control (P-control) and proportional-integral control (PI-control) strategies for collective transport by a team of point-mass robots with the same assumptions. We proved that the closed-loop system comprised of the payload and robots is exponentially stable with P-control and asymptotically stable with PI-control. We also analyzed the system's rate of convergence to the target velocity in the case of P-control, finding that it is mainly affected by the robots' distribution around the load and that it influences the load's total rotation and drift from the target path during the transient phase of motion. Our simulations verified the correctness of our analysis for different robot distributions around the load, as well as the effectiveness of the PI-control in driving the payload's motion to the desired path. A possible extension of this work is to include noise in the robots' velocity measurements and derive conditions on the controller gains that guarantee the robustness of the system's performance.

In Chapter 2, we also presented a decentralized adaptive control strategy for collective transport by nonholonomic robots. The controllers did not require inter-robot communication, information about the payload dynamics and geometry, or knowledge of the number of robots in the transport team and their distribution around the payload. In addition, since the desired manifolds of motion were designed to be consistent with the system's holonomic constraints, the robots were not required to begin the transport task in any specific configuration. We validated this control strategy in simulations with 3D models of robots with realistic dynamics. The simulations demonstrated the effectiveness of the strategy at driving the transport team to a target speed in a desired direction. One possible direction for future work is to consider transport teams in which the robots have manipulators with more degrees of freedom and modify the controller to achieve internal force regulation and payload transport along curved reference trajectories. Another direction for future work is to consider robots with force sensing capabilities and utilize their force measurements of the transport team's exertions on the payload, which constitutes a type of implicit communication through the payload, in feedback controllers that drive the payload to a time-varying target velocity.

In Chapter 3, we proposed an obstacle avoidance controller for a holonomic finite-dimensional robot in an unbounded, GPS-denied environment with unknown strictly convex obstacles. The controller relied only on the robot's local measurements and did not require any information about the locations and geometry of the obstacles. We first studied the case where the environment has a single obstacle and proved that with the proposed controller, no collision takes place and the robot converges to the desired velocity after it passes the obstacle. For the case of multiple obstacles, we proposed a switching control scheme and showed that the robot avoids collisions and converges to the target velocity if it uses the controller designed for the single-obstacle case for the closest obstacle at each time instant. Moreover, the robot never becomes trapped between any pair of obstacles (i.e., there are no local stable equilibrium points) if it uses a sufficiently small p in the equation for the virtual potential field, which is used in the repulsive term of the controller. We also combined the proposed controller with the PI controller in Chapter 2 to design a controller for velocity control of collective transport in an unbounded domain with strictly convex obstacles. We presented simulations of collective transport with this controller and left the theoretical analysis of the controller for future work.

In Chapter 4, we proposed a decentralized PD control strategy for a team of identical point-mass robots to collectively transport a payload to a target position. The controller only required the robots' local measurements and did not rely on predefined trajectories or explicit communication between the robots. We proved that with the proposed controller, the robots drive the payload to a neighborhood of the destination, where the steady-state distance between the payload's center of mass and the target position was only a function of the number of the robots and their distribution around the payload. A version of the proposed controller for three-dimensional collective transport tasks in space applications was recently developed in (**Farivarnejad, H.** *et al.*, 2021).

In Chapter 5, we first proposed a controller for obstacle avoidance in a bounded convex domain that contains convex obstacles. We considered single integrator dynamics for the robot and proved the robot's convergence to the destination. We also proved that the robot never collides with the obstacles and never get stuck between them. We validated our theoretical results with simulations in MATLAB and implementation on a holonomic robot. We then modified the proposed controller for position control of collective transport. We added a velocity dissipation term to design a decentralized control law for point-mass robots with double-integrator dynamics that perform collective transport in a bounded convex domain with convex obstacles. We proved the convergence of the system to a set of equilibrium points. We also validated the effectiveness of the proposed controller with simulations in MATLAB.

An additional direction for future work is to derive conditions that theoretically guarantee the absence of local equilibrium points for collective transport in bounded domains. We can also explore the application of extremum seeking to the design of controllers for collective transport tasks in GPS-denied bounded environments that contain both concave and convex obstacles.

REFERENCES

Ames, A. D., X. Xu, J. W. Grizzle and P. Tabuada, “Control barrier function based quadratic programs for safety critical systems”, IEEE Transactions on Automatic Control **62**, 8, 3861–3876 (2017).

Arslan, O. and D. E. Koditschek, “Sensor-based reactive navigation in unknown convex sphere worlds”, The International Journal of Robotics Research **38**, 2-3, 196–223 (2019).

Bai, H. and J. T. Wen, “Motion coordination through cooperative payload transport”, in “Proc. American Control Conference (ACC)”, pp. 1310–1315 (2009).

Bai, H. and J. T. Wen, “Cooperative load transport: A formation-control perspective.”, *IEEE Transactions on Robotics* **26**, 4, 742–750 (2010).

Bais, A. Z., S. Erhart, L. Zaccarian and S. Hirche, “Dynamic load distribution in cooperative manipulation tasks”, in “Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on”, pp. 2380–2385 (IEEE, 2015).

Belta, C., V. Isler and G. J. Pappas, “Discrete abstractions for robot motion planning and control in polygonal environments”, IEEE Transactions on Robotics **21**, 5, 864–874 (2005).

Berman, S., Q. Lindsey, M. S. Sakar, V. Kumar and S. Pratt, “Study of group food retrieval by ants as a model for multi-robot collective transport strategies”, in “Robotics: Science and Systems (RSS)”, (2010).

Bertsekas, D. P., A. Nedi and A. E. Ozdaglar, *Convex analysis and optimization* (Athena Scientific, 2003).

Bloch, A. M. *et al.*, *Nonholonomic mechanics and control* (Springer,, 2003).

Bräunl, T., “Omni-directional robots”, Embedded Robotics: Mobile Robot Design and Applications with Embedded Systems pp. 147–156 (2008).

Chen, J., M. Gauci, W. Li, A. Kolling and R. Groß, “Occlusion-based cooperative transport with a swarm of miniature mobile robots”, *IEEE Transactions on Robotics* **31**, 2, 307–321 (2015).

Conner, D. C., A. A. Rizzi and H. Choset, “Composition of local potential functions for global robot control and navigation”, in “Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems”, pp. 3546–3551 (2003).

Connolly, C. I., J. B. Burns and R. Weiss, “Path planning using Laplace’s equation”, in “Proceedings of the 1990 IEEE International Conference on Robotics and Automation”, pp. 2102–2106 (1990).

Culbertson, P. and M. Schwager, “Decentralized adaptive control for collaborative manipulation”, in “Accepted to the IEEE Conference on Robotics and Automation”, (2018).

Czaczkes, T. J. and F. L. Ratnieks, “Cooperative transport in ants (hymenoptera: Formicidae) and elsewhere”, Myrmecological News **18**, 1–11 (2013).

Dai, G. B. and Y. C. Liu, “Distributed coordination and cooperation control for networked mobile manipulators”, IEEE Transactions on Industrial Electronics **64**, 6, 5065–5074 (2017).

Ding, G., J. J. Koh, K. Merckaert, B. Vanderborght, M. M. Nicotra, C. Heckman, A. Roncone and L. Chen, “Distributed reinforcement learning for cooperative multi-robot object manipulation”, arXiv preprint arXiv:2003.09540 (2020).

Dohmann, P. B. G. and S. Hirche, “Distributed control for cooperative manipulation with event-triggered communication”, IEEE Transactions on Robotics (2020).

Erhart, S. and S. Hirche, “Model and analysis of the interaction dynamics in cooperative manipulation tasks”, *IEEE Transactions on Robotics* **32**, 3, 672–683 (2016).

Ferro, M., A. Paolillo, A. Cherubini and M. Vendittelli, “Vision-based navigation of omnidirectional mobile robots”, IEEE Robotics and Automation Letters **4**, 3, 2691–2698 (2019).

Filippidis, I. and K. J. Kyriakopoulos, “Adjustable navigation functions for unknown sphere worlds”, in “Proceedings of the 2011 IEEE International Conference on Decision and Control and European Control Conference”, pp. 4276–4281 (2011).

Fox, D., W. Burgard and S. Thrun, “The dynamic window approach to collision avoidance”, IEEE Robotics and Automation Magazine **4**, 1, 23–33 (1997).

Franchi, A., A. Petitti and A. Rizzo, “Distributed estimation of the inertial parameters of an unknown load via multi-robot manipulation”, in “53rd IEEE Conference on Decision and Control”, pp. 6111–6116 (2014).

Franklin, G. F., J. D. Powell and A. Emami-Naeini, *Feedback control of dynamic systems* (Prentice Hall Press, 2014).

Ge, S. and Y. Cui, “Dynamic motion planning for mobile robots using potential field method”, Autonomous Robots **13**, 3, 207–222 (2002).

Ge, S. S. and Y. J. Cui, “New potential functions for mobile robot path planning”, IEEE Transactions on Robotics and Automation **16**, 5, 615–620 (2000).

Gelblum, A., I. Pinkoviezky, E. Fonio, N. S. Gov and O. Feinerman, “Emergent oscillations assist obstacle negotiation during ant cooperative transport”, Proceedings of the National Academy of Sciences **113**, 51, 14615–14620 (2016).

Gueaieb, W., F. Karray and S. Al-Sharhan, “A robust adaptive fuzzy position/force control scheme for cooperative manipulators”, IEEE Transactions on Control Systems Technology **11**, 4, 516–528 (2003).

Guldner, J. and V. I. Utkin, “Sliding mode control for gradient tracking and robot navigation using artificial potential fields”, IEEE Transactions on Robotics and Automation **11**, 2, 247–254 (1995).

Habibi, G., K. Zachary, W. Xie, M. Jellins and J. McLurkin, "Distributed centroid estimation and motion controllers for collective transport by multi-robot systems", in "Proc. IEEE Int'l. Conf. on Robotics and Automation (ICRA)", (2015).

Ho, H. W., G. C. de Croon and Q. Chu, "Distance and velocity estimation using optical flow from a monocular camera", International Journal of Micro Air Vehicles **9**, 3, 198–208 (2017).

Horn, R. and C. Johnson, *Matrix Analysis*, Matrix Analysis (Cambridge University Press, 2012).

Ivanjko, E., T. Petrinic and I. Petrovic, "Modelling of mobile robot dynamics", in "7th EUROSIM Congress on Modelling and Simulation", vol. 2 (2010).

Jin, L., S. Li, X. Luo, Y. Li and B. Qin, "Neural dynamics for cooperative control of redundant robot manipulators", IEEE Transactions on Industrial Informatics **PP**, 99, 1–1 (2018).

Kalat, S. T., S. G. Faal and C. D. Onal, "A decentralized, communication-free force distribution method with application to collective object manipulation", Journal of Dynamic Systems, Measurement, and Control **140**, 9, 091012 (2018).

Karaman, S. and E. Frazzoli, "Sampling-based algorithms for optimal motion planning", The International Journal of Robotics Research **30**, 7, 846–894 (2011).

Khalil, H. K., *Nonlinear Systems* (Prentice Hall, Upper Saddle River, N.J., 1996).

Khatib, O., "Real-time obstacle avoidance for manipulators and mobile robots", The International Journal of Robotics Research **5**, 1, 90–98 (1986).

Kim, J.-O. and P. K. Khosla, "Real-time obstacle avoidance using harmonic potential functions", IEEE Transactions on Robotics and Automation **8**, 3, 338–349 (1992).

Kim, S., H. Seo, J. Shin and H. J. Kim, "Cooperative aerial manipulation using multirotors with multi-dof robotic arms", IEEE/ASME Transactions on Mechatronics **PP**, 99, 1–1 (2018).

Koditschek, D. E., "The application of total energy as a lyapunov function for mechanical control systems", Contemporary mathematics **97**, 131 (1989).

Koditschek, D. E. and E. Rimon, "Robot navigation functions on manifolds with boundary", Advances in Applied Mathematics **11**, 4, 412–442 (1990).

Lafmejani, A. S., **Farivarnejad, H.** and S. Berman, "Adaptation of gradient-based navigation control for holonomic robots to nonholonomic robots", Accepted to IEEE Robotics and Automation Letters (2020).

LaValle, S. M., *Planning algorithms* (Cambridge University Press, 2006).

Lee, H., H. Kim and H. J. Kim, "Planning and control for collision-free cooperative aerial transportation", IEEE Transactions on Automation Science and Engineering **PP**, 99, 1–13 (2017).

Lee, H., H. Kim, W. Kim and H. J. Kim, “An integrated framework for cooperative aerial manipulators in unknown environments”, IEEE Robotics and Automation Letters **PP**, 99, 1–1 (2018).

Li, C. and H. G. Tanner, “Navigation functions with time-varying destination manifolds in star worlds”, IEEE Transactions on Robotics **35**, 1, 35–48 (2019).

Li, X., Z. Xu, S. Li, H. Wu and X. Zhou, “Cooperative kinematic control for multiple redundant manipulators under partially known information using recurrent neural network”, IEEE Access **8**, 40029–40038 (2020).

Li, Z., C. Yang, C. Y. Su, S. Deng, F. Sun and W. Zhang, “Decentralized fuzzy control of multiple cooperating robotic manipulators with impedance interaction”, IEEE Transactions on Fuzzy Systems **23**, 4, 1044–1056 (2015).

Liberzon, D., *Switching in systems and control* (Springer Science & Business Media, 2003).

Lindemann, S. R. and S. M. LaValle, “Simple and efficient algorithms for computing smooth, collision-free feedback laws over given cell decompositions”, The International Journal of Robotics Research **28**, 5, 600–621 (2009).

Marino, A., “Distributed adaptive control of networked cooperative mobile manipulators”, IEEE Transactions on Control Systems Technology **PP**, 99, 1–15 (2017).

Marino, A., G. Muscio and F. Pierri, “Distributed cooperative object parameter estimation and manipulation without explicit communication”, in “2017 IEEE International Conference on Robotics and Automation (ICRA)”, pp. 2110–21116 (2017).

Marino, A. and F. Pierri, “A two stage approach for distributed cooperative manipulation of an unknown object without explicit communication and unknown number of robots”, Robotics and Autonomous Systems **103**, 122 – 133 (2018).

McCreery, H. F. and M. Breed, “Cooperative transport in ants: a review of proximate mechanisms”, Insectes Sociaux **61**, 2, 99–110 (2014).

Medina, O., S. Hacohen and N. Shvalb, “Robotic swarm motion planning for load carrying and manipulating”, IEEE Access **8**, 53141–53150 (2020).

Meriam, J. L. and L. G. Kraige, *Engineering mechanics: dynamics*, vol. 2 (John Wiley & Sons, 2012).

Michel, O., “Webots: Professional mobile robot simulation”, International Journal of Advanced Robotic Systems **1**, 1, 39–42 (2004).

Murray, R. M., S. S. Sastry and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation* (CRC Press, Inc., Boca Raton, FL, USA, 1994), 1st edn.

Ogren, P. and N. E. Leonard, “A convergent dynamic window approach to obstacle avoidance”, IEEE Transactions on Robotics **21**, 2, 188–195 (2005).

Padhy, R. P., S. K. Choudhury, P. K. Sa and S. Bakshi, “Obstacle avoidance for unmanned aerial vehicles: Using visual features in unknown environments”, IEEE Consumer Electronics Magazine **8**, 3, 74–80 (2019).

Parra-Vega, V., A. Sanchez, C. Izaguirre, O. Garcia and F. Ruiz-Sanchez, “Toward aerial grasping and manipulation with multiple UAVs”, *Journal of Intelligent & Robotic Systems* **70**, 1-4, 575–593 (2013).

Paternain, S., D. E. Koditschek and A. Ribeiro, “Navigation functions for convex potentials in a space with convex obstacles”, IEEE Transactions on Automatic Control **63**, 9, 2944–2959 (2018).

Paternain, S. and A. Ribeiro, “Safe online navigation of convex potentials in spaces with convex obstacles”, in “2017 IEEE 56th Annual Conference on Decision and Control (CDC)”, pp. 2473–2478 (2017).

Pliego-Jimenez, J. and M. Arteaga-Perez, “On the adaptive control of cooperative robots with time-variant holonomic constraints”, International Journal of Adaptive Control and Signal Processing **31**, 8, 1217–1231, acs.2758 (2017).

Ponce-Hinestroza, A.-N., J.-A. Castro-Castro, H.-I. Guerrero-Reyes, V. Parra-Vega and E. Olgu, “Cooperative redundant omnidirectional mobile manipulators: Model-free decentralized integral sliding modes and passive velocity fields”, in “2016 IEEE International Conference on Robotics and Automation (ICRA)”, pp. 2375–2380 (IEEE, 2016).

Prajna, S., A. Jadbabaie and G. J. Pappas, “Stochastic safety verification using barrier certificates”, in “Proceedings of the 2004 IEEE International Conference on Decision and Control”, pp. 929–934 (2004).

Ramírez-Llanos, E. and S. Martínez, “Stochastic source seeking for mobile robots in obstacle environments via the SPSA method”, IEEE Transactions on Automatic Control **64**, 4, 1732–1739 (2019).

Rassameepaiboon, W. and W. Assawinchaichote, “Vehicle avoidance reaction by two step motion flow cluster”, in “2018 22nd International Computer Science and Engineering Conference (ICSEC)”, pp. 1–6 (2018).

Rimon, E. and D. E. Koditschek, “Exact robot navigation using artificial potential functions”, IEEE Transactions on Robotics and Automation **8**, 5, 501–518 (1992).

Rubenstein, M., A. Cabrera, J. Werfel, G. Habibi, J. McLurkin and R. Nagpal, “Collective transport of complex objects by simple robots: Theory and experiments”, in “Proc. Int’l. Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS)”, pp. 47–54 (2013).

Sadati, N. and A. Ghaffarkhah, “Decentralized position and force control of nonredundant multi-manipulator systems”, in “2007 International Conference on Control, Automation and Systems”, pp. 2223–2229 (2007).

Sakai, D., H. Fukushima and F. Matsuno, “Flocking for multirobots without distinguishing robots and obstacles”, IEEE Transactions on Control Systems Technology **25**, 3, 1019–1027 (2017).

Shahidi, R., M. Shayman and P. S. Krishnaprasad, “Mobile robot navigation using potential functions”, in “Proceedings of the 1991 IEEE International Conference on Robotics and Automation”, pp. 2047–2053 (1991).

Shahrokhi, S. and A. T. Becker, “Object manipulation and position control using a swarm with global inputs”, in “2016 IEEE International Conference on Automation Science and Engineering (CASE)”, pp. 561–566 (2016).

Slotine, J.-J. E. and W. Li, *Applied Nonlinear Control* (Prentice Hall, Englewood Cliffs, N.J., 1991).

Tanner, H. G., A. Jadbabaie and G. J. Pappas, “Stable flocking of mobile agents, part i: fixed topology”, in “42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)”, vol. 2, pp. 2010–2015 Vol.2 (2003).

Farivarnejad, H. and S. Berman, “Stability and convergence analysis of a decentralized proportional-integral control strategy for collective transport”, in “2018 Annual American Control Conference (ACC)”, pp. 2794–2801 (2018).

Farivarnejad, H. and S. Berman, “Decentralized collective transport along manifolds compatible with holonomic constraints by robots with minimal global information”, in “Proceedings of the 2020 American Control Conference (ACC)”, pp. 2068–2075 (2020a).

Farivarnejad, H. and S. Berman, “Decentralized PD control for multi-robot collective transport to a target location using minimal information”, in “Unmanned Systems Technology XXII”, vol. 11425, pp. 38 – 48, International Society for Optics and Photonics (SPIE, 2020b).

Farivarnejad, H. and S. Berman, “Design and analysis of a potential-based controller for safe robot navigation in unknown gps-denied environments with strictly convex obstacles”, Systems & Control Letters **144**, 104772 (2020c).

Farivarnejad, H. and S. Berman, “On the formation of local minima in gradient-based decentralized control of collective transport in bounded convex domains”, In preparation for submission to the 60th IEEE Conference on Decision and Control, Austin, TX (2021).

Farivarnejad, H., A. S. Lafmejani, and S. Berman, “Fully decentralized controller for multi-robot collective transport in space applications”, Submitted to the IEEE Aerospace Conference, Big Sky, MT (2021).

Farivarnejad, H., A. S. Lafmejani and S. Berman, “Switching local navigation-like functions for safe robot navigation in a bounded domain with unknown convex obstacles”, In preparation for submission to IEEE Transactions on Robotics (2020).

Farivarnejad, H., S. Wilson and S. Berman, "Decentralized sliding mode control for autonomous collective transport by multi-robot systems", in "2016 IEEE 55th Conference on Decision and Control (CDC)", pp. 1826–1833 (2016).

Tsiamis, A., C. K. Verginis, C. P. Bechlioulis and K. J. Kyriakopoulos, "Cooperative manipulation exploiting only implicit communication", in "2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)", pp. 864–869 (2015).

Wang, L., A. D. Ames and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems", IEEE Transactions on Robotics **33**, 3, 661–674 (2017).

Wang, Z. and M. Schwager, "Multi-robot manipulation without communication", in "Proc. Int'l. Symposium on Distributed Autonomous Robotic Systems (DARS)", pp. 43–56 (2014).

Wang, Z. and M. Schwager, "Multi-robot manipulation with no communication using only local measurements", in "Proc. IEEE Conf. on Decision and Control (CDC)", (2015).

Wang, Z. and M. Schwager, "Kinematic multi-robot manipulation with no communication using force feedback", in "Proc. IEEE Int'l. Conf. on Robotics and Automation (ICRA)", (2016).

Wilson, S., R. Gameros, M. Sheely, M. Lin, K. Dover, R. Gevorkyan, M. Haberland, A. Bertozzi and S. Berman, "Pheeno, a versatile swarm robotic research and education platform", *IEEE Robotics and Automation Letters* **1**, 2, 884–891 (2016).

Xi, W., X. Tan and J. S. Baras, "A hybrid scheme for distributed control of autonomous swarms", in "Proceedings of the 2005 American Control Conference", pp. 3486–3491 (2005).

Yagiz, N., Y. Hacioglu and Y. Z. Arslan, "Load transportation by dual arm robot using sliding mode control", *Journal of Mechanical Science and Technology* **24**, 5, 1177–1184 (2010).

Yim, W., M. Selvarajan and W. R. Wells, "Sliding mode cooperative motion control of dual arm manipulators", *Artificial Life and Robotics* **3**, 3, 166–169 (1999).

Yufka, A. and M. Ozkan, "Formation-based control scheme for cooperative transportation by multiple mobile robots", International Journal of Advanced Robotic Systems **12** (2015).

APPENDIX A

UNCONSTRAINED DYNAMICS OF A NONHOLONOMIC ROBOT

Here, we derive the unconstrained dynamical model of a nonholonomic robot. Using the classical Lagrange formulation, we first obtain the constrained dynamics of the robot, and then eliminate the Lagrange multipliers in this constrained model. For simplicity, we drop the subscript i in the variables associated with the robot.

We begin with the vector of generalized coordinates $\boldsymbol{q}_c := [x \;\; y \;\; \theta \;\; \theta_R \;\; \theta_L]^T \in \mathbb{R}^5$, which completely describes the position of each point on the robot's core as it moves. We define m_c as the mass of the robot's core, I_c as the robot core's moment of inertia about the axis that passes through its center of mass and is normal to the plane of motion, and J_w as the moment of inertia of each wheel about its axis of rotation. Hence, the Lagrangian of the robot's core is written as:

$$\mathcal{L} = \frac{1}{2}m_c(\dot{x}^2 + \dot{y}^2) + \frac{1}{2}I_c\dot{\theta}^2 + \frac{1}{2}(J_w\dot{\theta}_R^2 + J_w\dot{\theta}_L^2). \tag{A.1}$$

Furthermore, to satisfy the rolling condition for each wheel, we include four constraint equations that can be written in the following matrix form:

$$\boldsymbol{A}_c\dot{\boldsymbol{q}}_c = \boldsymbol{0}, \tag{A.2}$$

where $\boldsymbol{A}_c \in \mathbb{R}^{4\times 5}$ is

$$\boldsymbol{A}_c = \begin{bmatrix} 1 & 0 & b\cos(\theta) & -r\cos(\theta) & 0 \\ 0 & 1 & b\sin(\theta) & -r\sin(\theta) & 0 \\ 1 & 0 & -b\cos(\theta) & 0 & -r\cos(\theta) \\ 0 & 1 & -b\sin(\theta) & 0 & -r\sin(\theta) \end{bmatrix}. \tag{A.3}$$

Defining $\boldsymbol{\lambda} := [\lambda_1 \; \lambda_2 \; \lambda_3 \; \lambda_4]^T \in \mathbb{R}^4$ as the vector of Lagrange multipliers, and using the Lagrange formulation, the equations of motion of the robot's core are calculated as:

$$\begin{aligned} m_c\ddot{x} &= \lambda_1 + \lambda_3 \\ m_c\ddot{y} &= \lambda_2 + \lambda_4 \\ I_c\ddot{\theta} &= \lambda_1 b\cos(\theta) + \lambda_2 b\sin(\theta) - \lambda_3 b\cos(\theta) - \lambda_4 b\sin(\theta) \\ J_w\ddot{\theta}_R &= \tau_R - \lambda_1 r\cos(\theta) - \lambda_2 r\sin(\theta) \\ J_w\ddot{\theta}_L &= \tau_L - \lambda_3 r\cos(\theta) - \lambda_4 r\sin(\theta), \end{aligned} \tag{A.4}$$

where τ_R and τ_L are the actuation torques on the right and left wheels, respectively.

Using a similar approach to the method in Section 1.4 of Bloch *et al.* (2003) for deriving the nonholonomic dynamics of a vertical rolling disk, we eliminate the Lagrange multipliers from these equations to obtain the unconstrained equations of motion for the robot's core. Since $rank(\boldsymbol{A}_c) = 3$, the four constraint equations A.2 are linearly dependent, and the number of linearly independent constraints is 3. These three constraints can be calculated from basic row operations on the matrix $\boldsymbol{A}_c$ in Eq. A.3. By adding the first and third rows and the second and fourth rows of $\boldsymbol{A}_c$, we obtain the following two equations from Eq. A.2:

$$\dot{x} = \frac{r}{2}(\dot{\theta}_R + \dot{\theta}_L)\cos(\theta)\ , \quad \dot{y} = \frac{r}{2}(\dot{\theta}_R + \dot{\theta}_L)\sin(\theta). \tag{A.5}$$

Subtracting the third row of $\boldsymbol{A}_c$ from the first row (or the fourth row from the second row) yields the third equation:

$$\dot{\theta} = \frac{r}{2b}(\dot{\theta}_R - \dot{\theta}_L). \tag{A.6}$$

Eq. A.5–Eq. A.6 are the three linearly independent constraint equations. Differentiating these equations with respect to time, and substituting the resulting expressions for $\ddot{x}$, $\ddot{y}$, and $\ddot{\theta}$ into the first, second, and third equations in Eq. A.4, we obtain:

$$\begin{aligned}
&\frac{r}{2}\left(\cos(\theta)(\ddot{\theta}_R + \ddot{\theta}_L) - \dot{\theta}\sin(\theta)(\dot{\theta}_R + \dot{\theta}_L)\right) = \frac{1}{m_c}(\lambda_1 + \lambda_3)\\
&\frac{r}{2}\left(\sin(\theta)(\ddot{\theta}_R + \ddot{\theta}_L) + \dot{\theta}\cos(\theta)(\dot{\theta}_R + \dot{\theta}_L)\right) = \frac{1}{m_c}(\lambda_2 + \lambda_4)\\
&\frac{r}{2b}I_c(\ddot{\theta}_R - \ddot{\theta}_L) = b\cos(\theta)(\lambda_1 - \lambda_3) + b\sin(\theta)(\lambda_2 - \lambda_4).
\end{aligned} \tag{A.7}$$

Adding the fourth and fifth equations in Eq. A.4 results in the equation:

$$\begin{aligned}
J_w(\ddot{\theta}_R + \ddot{\theta}_L) = {} & (\tau_R + \tau_L) - r\cos(\theta)(\lambda_1 + \lambda_3)\\
& - r\sin(\theta)(\lambda_2 + \lambda_4).
\end{aligned} \tag{A.8}$$

By substituting in the expressions for $\lambda_1 + \lambda_3$ and $\lambda_2 + \lambda_4$ from Eq. A.7, we obtain:

$$\left(J_w + \frac{m_c r^2}{2}\right)(\ddot{\theta}_R + \ddot{\theta}_L) = \tau_R + \tau_L. \tag{A.9}$$

Subtracting the fifth equation from the fourth equation in Eq. A.4, we have:

$$\begin{aligned}
J_w(\ddot{\theta}_R - \ddot{\theta}_L) = {} & (\tau_R - \tau_L) - r\cos(\theta)(\lambda_1 - \lambda_3)\\
& - r\sin(\theta)(\lambda_2 - \lambda_4).
\end{aligned} \tag{A.10}$$

If we move the term $(\tau_R - \tau_L)$ to the left-hand side of the above equation, then the right-hand side is equal to the right-hand side of the third equation in Eq. A.7 multiplied by $-r/b$. Therefore, Eq. A.10 can be rewritten as:

$$\left(J_w + \frac{r^2}{2b^2}I_c\right)(\ddot{\theta}_R - \ddot{\theta}_L) = \tau_R - \tau_L. \tag{A.11}$$

The Lagrange multipliers, *i.e.* the elements of $\boldsymbol{\lambda}$, have been eliminated in Eq. A.9 and Eq. A.11. Also, since the number of original generalized coordinates is 5 and the number of linearly independent constraints is 3, the robot's core has only 2 degrees of freedom, and the unconstrained dynamics of the core are therefore expressed by two equations. Hence, Eq. A.9 and Eq. A.11 are the unconstrained equations of motion for the robot's core. Finally, defining

$$H_1 = J_w + \frac{m_c r^2}{2}, \quad H_2 = J_w + \frac{r^2}{2b^2}I_c, \tag{A.12}$$

we can write Eq. A.9 and Eq. A.11 in matrix form:

$$\begin{bmatrix} H_1 & H_1 \\ H_2 & -H_2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_R \\ \ddot{\theta}_L \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix} . \tag{A.13}$$

By pre-multiplying this equation by the inverse of the matrix that multiplies the vector $[\tau_R \; \tau_L]^T$, it can be rewritten as

$$\frac{1}{2} \begin{bmatrix} H_1 + H_2 & H_1 - H_2 \\ H_1 - H_2 & H1 + H_2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_R \\ \ddot{\theta}_L \end{bmatrix} = \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix} , \tag{A.14}$$

which is in the standard form of unconstrained dynamics. This formulation shows that $\boldsymbol{q}_c^* = [\theta_R \; \theta_L]^T \in \mathbb{R}^2$ is an unconstrained configuration space for the dynamics of a nonholonomic robot.

APPENDIX B

CALCULATION OF THE GRADIENTS OF φ WITH RESPECT TO $\boldsymbol{d}$ AND $\boldsymbol{q}$

We can write the gradient of φ with respect to $\boldsymbol{d}$ as

$$\boldsymbol{\nabla}_{\boldsymbol{d}}\varphi = \frac{\partial \varphi}{\partial \boldsymbol{d}} = \frac{\partial \varphi}{\partial \delta}\frac{\partial \delta}{\partial \boldsymbol{d}}. \tag{B.1}$$

We represent $\boldsymbol{d}$ in terms of its components in the global coordinate frame as $\boldsymbol{d} := [d_x \; d_y]^T$. Then, from the definition of δ, we have that $\delta = ||\boldsymbol{d}|| - r = \sqrt{d_x^2 + d_y^2} - r$. Using this expression for δ, we obtain:

$$\frac{\partial \delta}{\partial \boldsymbol{d}} = \begin{bmatrix} \frac{\partial \delta}{\partial d_x} \\ \frac{\partial \delta}{\partial d_y} \end{bmatrix} = \begin{bmatrix} \frac{d_x}{\sqrt{d_x^2+d_y^2}} \\ \frac{d_y}{\sqrt{d_x^2+d_y^2}} \end{bmatrix} = \frac{1}{\sqrt{d_x^2 + d_y^2}} \begin{bmatrix} d_x \\ d_y \end{bmatrix} = \boldsymbol{e_d}. \tag{B.2}$$

Therefore, $\boldsymbol{\nabla}_{\boldsymbol{d}}\varphi$ is given by

$$\boldsymbol{\nabla}_{\boldsymbol{d}}\varphi = \frac{\partial \varphi}{\partial \boldsymbol{d}} = \frac{\partial \varphi}{\partial \delta}\boldsymbol{e_d}. \tag{B.3}$$

We also represent the position of the projection point in terms of its components in the global frame as $\boldsymbol{q}_P = [q_{p,x} \; q_{p,y}]^T$. Then the vector equation $\boldsymbol{q} = \boldsymbol{d} + \boldsymbol{q}_P$ can be written as

$$d_x = x - q_{p,x} \;, \quad d_y = y - q_{p,y}. \tag{B.4}$$

The gradient of φ with respect to $\boldsymbol{q}$ can be calculated as

$$\boldsymbol{\nabla}_{\boldsymbol{q}}\varphi = \frac{\partial \varphi}{\partial \boldsymbol{q}} = \frac{\partial \varphi}{\partial \delta}\frac{\partial \delta}{\partial \boldsymbol{q}}. \tag{B.5}$$

By the chain rule, the term $\frac{\partial \delta}{\partial \boldsymbol{q}}$ can be expressed as

$$\frac{\partial \delta}{\partial \boldsymbol{q}} = \frac{\partial \delta}{\partial d_x}\frac{\partial d_x}{\partial \boldsymbol{q}} + \frac{\partial \delta}{\partial d_y}\frac{\partial d_y}{\partial \boldsymbol{q}}, \tag{B.6}$$

which can be rewritten as

$$\frac{\partial \delta}{\partial \boldsymbol{q}} = \frac{\partial \delta}{\partial d_x}\begin{bmatrix} \frac{\partial d_x}{\partial x} \\ \frac{\partial d_x}{\partial y} \end{bmatrix} + \frac{\partial \delta}{\partial d_y}\begin{bmatrix} \frac{\partial d_y}{\partial x} \\ \frac{\partial d_y}{\partial y} \end{bmatrix}. \tag{B.7}$$

Given Eq. B.4 and the fact that $\delta = \sqrt{d_x^2 + d_y^2} - r$, we can calculate the partial derivatives in Eq. B.7 to obtain:

$$\frac{\partial \delta}{\partial \boldsymbol{q}} = \frac{d_x}{\sqrt{d_x^2 + d_y^2}}\begin{bmatrix} 1 \\ 0 \end{bmatrix} + \frac{d_y}{\sqrt{d_x^2 + d_y^2}}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{d_x^2 + d_y^2}}\begin{bmatrix} d_x \\ d_y \end{bmatrix} = \boldsymbol{e_d}. \tag{B.8}$$

Substituting this expression for $\frac{\partial \delta}{\partial \boldsymbol{q}}$ into Eq. B.5, we find that

$$\boldsymbol{\nabla}_{\boldsymbol{q}}\varphi = \frac{\partial \varphi}{\partial \delta}\boldsymbol{e_d}, \tag{B.9}$$

which is identical to Eq. B.3. Therefore, we conclude that

$$\boldsymbol{\nabla}_{\boldsymbol{d}}\varphi = \boldsymbol{\nabla}_{\boldsymbol{q}}\varphi. \tag{B.10}$$

APPENDIX C

PROOF OF LEMMA 3.3.12

We describe a procedure for choosing a and b in Eq. 3.28 in order to ensure a strictly positive lower bound γ on the time response $\varrho(t)$ of the system in Eq. 3.27. We know that any unforced scalar linear second-order system can be written in the following form (Franklin *et al.*, 2014),

$$\ddot{\varrho} + 2\zeta\omega_n\dot{\varrho} + \omega_n^2\varrho = 0. \tag{C.1}$$

Hence, the system in Eq. 3.27 can be represented as

$$\ddot{\varrho} + 2\zeta\omega_n\dot{\varrho} + \omega_n^2\varrho = b, \tag{C.2}$$

where

$$\omega_n = \sqrt{a}, \tag{C.3}$$

$$\zeta = \frac{k}{2\sqrt{a}}, \tag{C.4}$$

and consequently, its time response is written as

$$\varrho(t) = e^{-\zeta\omega_n t}\left(c_1\cos(\omega_n t) + c_2\sin(\omega_n t)\right) + b^{'}, \quad t \geq 0, \tag{C.5}$$

in which

$$c_1 \;=\; \varrho_0 - \frac{b}{\omega_n^2}, \tag{C.6}$$

$$c_2 \;=\; \frac{1}{\omega_n}(w_0 + c_1\zeta\omega_n), \tag{C.7}$$

$$b^{'} \;=\; \frac{b}{\omega_n^2}. \tag{C.8}$$

Let us choose b such that $c_1 = 0$ [1], i.e.,

$$b := \varrho_0\omega_n^2. \tag{C.9}$$

Then, from Eq. C.5, we obtain the following inequality:

$$\varrho(t) = e^{-\zeta\omega_n t}c_2\sin(\omega_n t) + b^{'} \geq -|c_2| + b^{'}, \quad t \geq 0. \tag{C.10}$$

In order to ensure that $\varrho(t) \geq \gamma$ for an arbitrary $\gamma \in (0, \varrho_0)$, we therefore need to enforce the condition $-|c_2| + b^{'} \geq \gamma$. To do this, we choose $b^{'}$ to satisfy this condition. Using Eq. C.7 with $c_1 = 0$, this condition can be written as:

$$b^{'} \;\geq\; |c_2| + \gamma = \frac{|w_0|}{\omega_n} + \gamma. \tag{C.11}$$

[1] This is not the only feasible choice for b. This is the most convenient choice that facilitates the calculations.

Taking into account the fact that $|w_0| \leq v_{max}$, we can ensure that Eq. C.11 is true by defining $b^{'}$ such that:

$$b^{'} \geq \frac{v_{max}}{\omega_n} + \gamma. \tag{C.12}$$

Noting that $b^{'} = \varrho_0$ from Eq. C.8 and Eq. C.9, the above inequality implies that

$$\omega_n \geq \frac{v_{max}}{(\varrho_0 - \gamma)}. \tag{C.13}$$

Since $a = \omega_n^2$ by Eq. C.3, Eq. C.13 implies that a should be chosen such that

$$a \geq \frac{v_{max}^2}{(\varrho_0 - \gamma)^2}. \tag{C.14}$$

By Eq. C.3 and Eq. C.9, we have that

$$b = \varrho_0 a. \tag{C.15}$$

We can then define b according to the selected value of a.

This proof shows that the establishment of a specific lower bound γ for the time response of the system in Eq. 3.27 requires a and b to be chosen such that the conditions in Eq. C.14 and Eq. C.15 hold. Also, the selection of sufficiently large values for a and b never contradicts Eq. 3.30, since we can always choose values of a and b such that the corresponding green line in Fig. 3.5 lies below the orange curve in that figure.