Cooperative Object Manipulation
A Rigidity Approach

Douglas Goldenberg

Cooperative Object Manipulation
A Rigidity Approach

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Aerospace engineering

Douglas Goldenberg

Submitted to the Senate
of the Technion — Israel Institute of Technology
Tishri 5779 Haifa October 2018

This research was carried out under the supervision of Associate Prof. Daniel Zelazo, at the Faculty of Aerospace Engineering.

This work was supported by the Israel Science Foundation under grant 1490/13.
The generous financial help of the Technion is gratefully acknowledged.

Contents

Abstract

Moving objects with a single robot is a common practice; however, the weight and dimensions of the cargo oftentimes limit the task. To aid this, we propose using multiple coordinating robots to move large objects. The main challenge in this approach is that the robots do not know the geometry of the object they are moving, but do have complete information regarding their own relative formation (that is, their position, orientation and velocity). This fact motivates solving the problem using rigidity theory, a tool recently used in formation control. Within this context, the planar motion of the object can be decoupled into pure steady state translations and rotations by utilizing the eigenvectors of the corresponding framework's rigidity matrix. This formulation is beneficial because it only requires local information to create any desired trajectory. In other words, utilizing the robots' position and velocity with respect to the system's center of mass, we are able to produce the forces needed for translations and rotations. This work explores how the rigidity matrix can be used to find the forces needed to shift and rotate an object, potentially allowing the robots to move that object anywhere in the plane. Finally, we demonstrate the analytical results with numerical simulations.

Abbreviations and Notations

FOC	:	Frame of coordinates.
CoM	:	Center of mass.
CoF	:	Centroid of the formation.
ss	:	Steady state.
i.c.	:	Initial conditions.

Chapter 1

Introduction and Motivation

Humans have been moving objects since the dawn of time. Whether it was hunters and gatherers bringing the daily catch back to the village, or mine workers carrying the ore to the closest market, the task remained the same. Nowadays our needs have changed and we no longer have to hunt for our meals, we simply go to the supermarket; but we still have to push the cart. We can also buy our products online, but we still have to go to the post office to get them. Since the very beginning we've been looking for solutions to our moving problems: getting animals to pull the plow in a field (which later on were replaced by tractors), or wagons in a railroad to carry coal, or drones to deliver our online purchases to our doorstep.

The main issue that we encounter when carrying payloads, is that to carry a large (or heavy) object, we need *another* large object. This may not be an issue in a construction site, where a crane is used to move tremendous weights, but is definitely an issue if we want a drone to deliver something heavy to a customer, or if the heavy object has to be transported indoors.

The solution we propose is not new in its robotic aspect, and it is based on human behavior. When we need to move a heavy piece of furniture at home (say, a bed), we don't use a forklift, we simply ask someone to help us. Without realizing it perhaps, what we're asking of that person is to share the load in a distributed manner. It is precisely this principle that guided us to the thesis of this work - *utilizing multiple robots to carry objects, instead of just one.*

This way of carrying things has many benefits. To begin with, we can use the same technique to move objects of all shapes and sizes, whereas forklifts for instance, are categorized by their maximal capacity. This means that a warehouse that up until today had to keep on site different (and expensive) forklifts could one day replace them with a lot of small (and cheap) robots that work simultaneously on moving multiple payloads, and collaborate when in need of moving something bigger. Another advantage is the reliability of the new approach. A malfunctioning robot can easily be replaced by another one in the fleet, while a malfunctioning crane brings construction to a halt. With this same example we can clearly see that this new solution is better from an economic point of view, but that is not all. The key advantage is the wide range of applications for this. In the future we could replace cranes, forklifts, trains, trucks and even planes with distributed carriers, where the only difference between the applications would be the number of carriers.

This work deals with the manipulation of a rigid body in the plane. In contrast to traditional ways to displace objects, where a single operator performs the task, we propose moving large objects with cooperating robots.

1.1 Literature Survey

To be able to analyze the dynamics of a rigid body [3] manipulated by multiple robots simultaneously, we need to understand the interaction between the robots involved. Graph theory [8] is a branch of Mathematics that can be used to model the interactions between these robots, by seeing them as the nodes of a graph and the passage of information between them as the edges. Another interpretation of a group of agents is a bar and joint framework, where the robots are the joints and the information between them is the bars. Such a framework can be either flexible or rigid, depending on the assembly. For instance in Figure 1.1a we can see a flexible framework. It is flexible because we can change its shape without changing the length of the bars. However in Figure 1.1b the framework is rigid, because even if the joints allow some motion of the bars, any force on them will result in either the motion of the entire framework, or in breaking the framework.

(a) A *flexible* bar and joint framework. (b) A *rigid* bar and joint framework.

Figure 1.1: Two simple bar and joint frameworks, flexible and rigid.

In our formulation the robots carrying the object are rigidly attached to it, therefore the framework that they represent should be modeled as rigid (otherwise the robots would be allowed motion *with respect to* the object, but this would mean that the robots move instead of the object). This representation is useful because we can transform the problem of having a group of robots push an object, into the problem of having a bar and joint framework [10] conserve its shape after forces are applied through the joints. To that end, we need to explore the rigidity properties of the framework, which have been thoroughly studied in [2], [4]. In particular, we're interested in the rigidity matrix and its relationship to the stresses of the framework, with the aim of establishing the connection between the two. Similar work has been done by [16], where the authors compute the eigenvectors of the symmetric rigidity matrix and apply them as velocities to the nodes of a graph.

Other research groups have demonstrated how to move objects without utilizing rigidity theory, such as in [17], where a leader applies a force to the payload and the rest of the robots align their forces accordingly. Another example is [11], where two robots use a depth camera to tilt an object onto wheels before pushing it to its destination. The authors in [5] utilize a group of cooperating robots to compute the mass, moment of inertia, center of mass (and more) of an object by applying forces to it, with little limitations on the formation's graph. In [15], an object is manipulated by two quadcopters by suspending the object from a cable connected to the cooperating robots. The authors of [13] enclose an object with 3 robots and plan the robots' paths such that the caged object is moved along a desired trajectory. In the same field, [1] shows how to move an object of polygonal shape by planning a series of pushes that are normal to the body's laterals. Another example of object manipulation by caging is described in [9], where using graph connectivity the formation of a group of robots is preserved while allowing individual tasks of the agents.

In general, the problem of moving an object by cooperating robots is not new, and some sides of it have been analyzed in [7] (for instance). The applications for such a task have also been presented before (see [12]), but this work differs from the existing studies, in the formulation of the dynamics, the generalization of the shape of the object to be moved, and the tools with which the task is completed.

1.2 Thesis Contribution and Outline

This work's main contributions can be summarized as follows:

- We find a connection between the forces required to move the object, and rigidity theory, by showing that the null space of the rigidity matrix is the image of the transposed motion matrix of the system.
- We fully analyze the dynamics of a rigid body moved by n cooperating robots.

This thesis is organized as follows. We start at Chapter 2 by covering the basic fields required to conduct this research, such as rigid body kinematics, graph and rigidity theory. In Chapter 3 the dynamics of the system are fully analyzed and the symmetric case is briefly covered analytically and in simulation. We then dictate the type of motion we are after (in the general, asymmetric case), and find the forces required to do so in Chapter 4. With those forces in hand we show that they can be computed via the rigidity matrix of the corresponding framework, and thus show the connection between this rigidity matrix and the motion matrix of the system (defined in this text). These forces are then tested in simulation and the results are corroborated in Chapter 5.

Chapter 2

Mathematical Background

In this chapter we'll cover the essentials of the mathematical fields used in this work. We begin with rigid body kinematics, which is needed to describe the velocity of different points on the rigid body the robots will attempt to move. We then introduce basic concepts from graph theory, used to describe the abstract sharing of information between agents in the system. Finally, notions from rigidity theory are used to find the connection between the rigidity matrix of the formation and the forces needed to move the object as intended.

This work employs standard mathematical notations. The n dimensional Euclidean space is denoted by $\mathbb{R}^n$, and $\mathbb{R}^{m\times n}$ is the set of $m \times n$ real matrices. For a matrix $A \in \mathbb{R}^{m\times n}$, $[A]_{ij}$ denotes the ijth entry of A. The null space of A is denoted as $Null\{A\}$. The image of the matrix is denoted as $Im\{A\}$. The $n \times n$ identity matrix is denoted I_n. The n-dimensional vector of all ones is denoted $\mathbb{1}_n = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T$. The Euclidean 2-norm of a vector $v \in \mathbb{R}^2$ is denoted $\|v\| = \sqrt{v^T v}$. Finally, for a time-varying signal $x(t)$, we denote its derivative as $\dot{x} = dx/dt$.

2.1 Rigid Body Kinematics and Planar Motion

Consider a rigid body in the plane, such as the one depicted in Figure 2.1. Points A and B are marked by the vectors $p_A, p_B \in \mathbb{R}^2$ respectively, with respect to the origin of a stationary, right hand frame of coordinates (FOC). The vector p_{BA} marks the position of point B with respect to point A, and can be written as $p_{BA} = p_B - p_A$.

Clearly if the line defined by the points A and B rotates about point A, then all the points in that line would have the same rotational velocity, say, $\dot{\theta} \in \mathbb{R}$. However, the same cannot be said about the linear velocities $\dot{p}_A, \dot{p}_B$. Instead, the connection between these can be written as follows:

$$\dot{p}_B = \dot{p}_A + \dot{\theta} T p_{BA}, \tag{2.1}$$

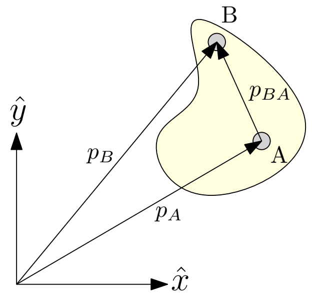


Figure 2.1: A representation of a rigid body with points A and B marked on it.

where T is the rotation matrix,

$$T = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}. \tag{2.2}$$

Next we develop the governing equations of motion of a rigid body under external forces. Assume a rigid body with mass and moment of inertia m and I, respectively. Let p_i be the point on the body where a force f_i is applied. The rigid body is also subject to a friction force at the point p_i, $F_i = -\mu m g \dot{p}_i$, where μ and g are the friction and acceleration constants. The angle θ marks the orientation of the body, as illustrated in Figure 2.2.

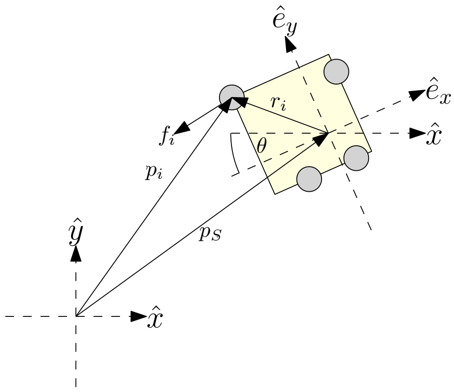


Figure 2.2: A rigid body moved by n forces f_i.

We consider n forces applied to the rigid body. The equations of motion for the center of mass of the body, denoted p_S, are then

$$m\ddot{p}_S = \sum_{i=1}^{n} (f_i + F_i) \tag{2.3}$$

$$\mathrm{I}\ddot{\theta} = \sum_{i=1}^{n} (f_i + F_i)^T T r_i. \tag{2.4}$$

Equivalently, in matrix form they can be expressed as,

$$m\ddot{p}_S = \left(\mathbb{1}_n^T \otimes I_2\right)(f + F) \tag{2.5}$$

$$\mathrm{I}\ddot{\theta} = (f + F)^T (I_n \otimes T)\, r, \tag{2.6}$$

where $\otimes$ stands for the *Kronecker Product* [18]. We denote the stacked vector of forces as $f = \begin{bmatrix} f_1^T & \cdots & f_n^T \end{bmatrix}^T$ and $F = \begin{bmatrix} F_1^T & \cdots & F_n^T \end{bmatrix}^T$. In addition, $r_i = p_i - p_S$ marks the position of point p_i with respect to point p_S,

$$r = \begin{bmatrix} r_1^T & \cdots & r_n^T \end{bmatrix}^T. \tag{2.7}$$

Substituting the expression for the friction forces into (2.5) gives

$$m\ddot{p}_S = \left(\mathbb{1}_n^T \otimes I_2\right)\left(f - \frac{\mu m g}{n}\dot{p}\right) \tag{2.8}$$

$$\mathrm{I}\ddot{\theta} = \left(f - \frac{\mu m g}{n}\dot{p}\right)^T (I_n \otimes T)\, r. \tag{2.9}$$

The linear velocity of the points p_i can be expressed in terms of the linear velocity of the center of mass as $\dot{p}_i = \dot{p}_S + \dot{\theta} T r_i$, leading to

$$\dot{p} = \mathbb{1}_n \otimes \dot{p}_S + \dot{\theta}\,(I_n \otimes T)\, r \tag{2.10}$$

and

$$m\ddot{p}_S = \left(\mathbb{1}_n^T \otimes I_2\right)\left\{f - \frac{\mu m g}{n}\left[\mathbb{1}_n \otimes \dot{p}_S + \dot{\theta}\,(I_n \otimes T)\, r\right]\right\} \tag{2.11}$$

$$\mathrm{I}\ddot{\theta} = \left\{f - \frac{\mu m g}{n}\left[\mathbb{1}_n \otimes \dot{p}_S + \dot{\theta}\,(I_n \otimes T)\, r\right]\right\}^T (I_n \otimes T)\, r. \tag{2.12}$$

After a bit more simplifications:

$$m\ddot{p}_S = \left(\mathbb{1}_n^T \otimes I_2\right) f - \mu m g\left(\dot{p}_S + \dot{\theta} T \bar{r}\right) \tag{2.13}$$

$$\mathrm{I}\ddot{\theta} = f^T (I_n \otimes T)\, r - \mu m g\left(\dot{p}_S^T T \bar{r} + \dot{\theta}\frac{r^T r}{n}\right), \tag{2.14}$$

where $\bar{r} = \frac{1}{n}\sum_{i=1}^{n} r_i$ is the average of the relative positions vector.

2.2 Graph Theory

A graph is the mathematical representation of the relation between pairs of objects. It is an abstract way to express the passage of information between agents in a network [8]. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the undirected graph defined by the vertex set $\mathcal{V} = \{1, \cdots, n\}$ and the edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, where n is the number of nodes in the graph, and ε is the number of edges. A graph can also be directed, simply by setting an orientation to the edges, that is, assigning a head and a tail to each edge. An orientation is useful to define the incidence matrix of the graph $H \in \mathbb{R}^{\varepsilon \times n}$, whose entries are determined as follows,

$$[H]_{ki} = \begin{cases} 1 & \text{node } i \text{ is the head of edge } k \\ -1 & \text{node } i \text{ is the tail of edge } k \\ 0 & \text{otherwise} \end{cases} . \tag{2.15}$$

If $(i, j) \in \mathcal{E}$ then nodes i and j are adjacent. The neighborhood of node i is then the set of all nodes that are adjacent to it,

$$\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}. \tag{2.16}$$

An example of a graph can be seen in Figure 2.3. In this example the node and edge sets are $\mathcal{V} = \{1, 2, 3, 4\}$ and $\mathcal{E} = \{(1, 2); (1, 3); (1, 4); (2, 3)\}$. Figures 2.3a and 2.3b show the difference between the directed and undirected graph, as well as the meaning of adjacency and neighborhood. In this case nodes 4 and 1 are adjacent, but 4 and 3 are not. In addition, the definitions of head and tail of edges can be seen in Figure 2.3b. Take for instance edge number 1 (marked in the figure simply as e_1): its tail is at node 1, and its head is at node 2. Here, the incidence matrix would be

$$H = \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & -1 & 1 & 0 \end{bmatrix} . \tag{2.17}$$

Note that $H\mathbb{1}_n = \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix}^T$. That is, $\mathbb{1}_n \in null\{H\}$ for any graph [6].

(a) A 4 noded *undirected* graph.

(b) A 4 noded *directed* graph.

Figure 2.3: The same graph (left) undirected and (right) directed.

2.3 Rigidity Theory

In this section we cover some topics of rigidity theory, which is a tool that we can use to characterize frameworks. Rigidity plays a major role in this research because it will help us find a shortcut to the forces we need to compute in order to move an object in a distributed manner.

A framework [10] $\mathcal{F} = (\mathcal{G}, P)$ is the pairing of a graph $\mathcal{G}$ and an embedding P onto a metric space (for instance $\mathbb{R}^2$). This means simply assigning a position in the plane to each of the nodes in the graph. Thus, $P(v_i) = p_i$ is the position of node i under the embedding P,

$$\begin{aligned} P &: \mathcal{V} \to \mathbb{R}^2 \\ v_i &\mapsto p_i = \begin{bmatrix} p_i^x & p_i^y \end{bmatrix}^T. \end{aligned} \tag{2.18}$$

Giving positions to the nodes of a graph is useful for comparing frameworks. In the example of Figure 2.4, the embedding takes the nodes to the positions:

$$p_1 = \begin{bmatrix} 2 \\ 4 \end{bmatrix}, \quad p_2 = \begin{bmatrix} 5 \\ 5 \end{bmatrix}, \quad p_3 = \begin{bmatrix} 6 \\ 2 \end{bmatrix}, \quad p_4 = \begin{bmatrix} 4 \\ 2 \end{bmatrix}, \tag{2.19}$$

or simply

$$p = \begin{bmatrix} p_1^T & p_2^T & p_3^T & p_4^T \end{bmatrix}^T = \begin{bmatrix} 2 & 4 & 5 & 5 & 6 & 2 & 4 & 2 \end{bmatrix}^T. \tag{2.20}$$

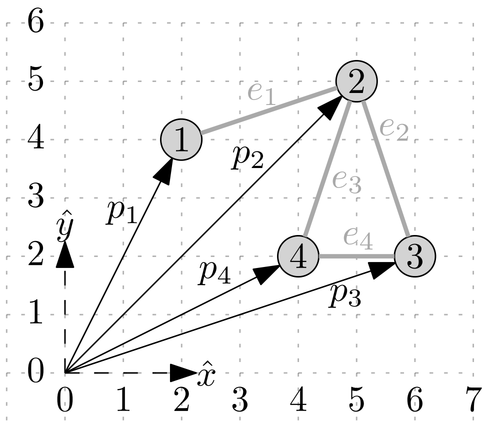


Figure 2.4: A framework where the position vectors are marked in black and the underlying graph with grey.

Next we mark the relative position of node i from node j (whom edge k connects) as

$$p_i - p_j = d_k, \tag{2.21}$$

such that the length of that edge is

$$\|p_{ij}\| = \|p_i - p_j\| = \|d_k\|. \tag{2.22}$$

In the example of Figure 2.4,

$$\begin{array}{cc} d_1 = p_{12} = \left[\begin{array}{cc} 3 & 1 \end{array}\right]^T & d_2 = p_{23} = \left[\begin{array}{cc} 1 & -3 \end{array}\right]^T \\ d_3 = p_{24} = \left[\begin{array}{cc} -1 & -3 \end{array}\right]^T & d_4 = p_{34} = \left[\begin{array}{cc} 2 & 0 \end{array}\right]^T. \end{array} \tag{2.23}$$

Note that we arbitrarily write

$$d_k = \begin{cases} p_{ij} & i < j \\ p_{ji} & i > j. \end{cases} \tag{2.24}$$

Referring to the lengths of edges gives us the ability to compare frameworks. This leads to the following notions of *equivalence* and *congruence* of frameworks.

Definition 2.3.1 (Equivalent frameworks)**.** Two frameworks $\mathcal{F}_1 = (\mathcal{G}, P)$ and $\mathcal{F}_2\,(\mathcal{G}, Q)$ are *equivalent* if

$$\|p_{ij}\| = \|q_{ij}\|\,\forall\,(i,j) \in \mathcal{E}. \tag{2.25}$$

Definition 2.3.2 (Congruent frameworks)**.** Two frameworks $\mathcal{F}_1 = (\mathcal{G}, P)$ and $\mathcal{F}_2\,(\mathcal{G}, Q)$ are *congruent* if

$$\|p_{ij}\| = \|q_{ij}\|\,\forall i, j \in \mathcal{V}. \tag{2.26}$$

In other words, two frameworks are equivalent if every edge has the same length in the two frameworks, and two frameworks are congruent if the distance between any two nodes (whether there is and edge connecting them or not) is the same in the two frameworks.

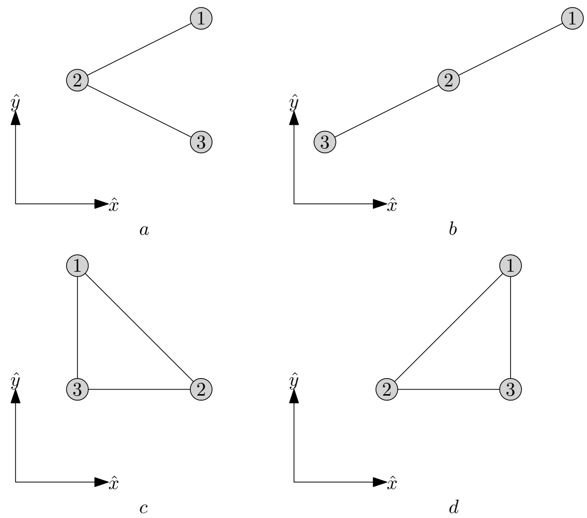


Figure 2.5: An example of 4 frameworks, where 2 of them are congruent, and 2 of them are just equivalent.

Figure 2.5 exhibits the difference between equivalency and congruency. Frameworks a and b are equivalent because the lengths of their edges are the same, however they are not congruent because the distance between nodes 1 and 3 is not the same. On the other hand, frameworks c and d are congruent because the distance between any two nodes is equal.

Another notion that should be discussed, is the possibility of having the nodes of the framework move over time, and even follow a predetermined path. We can imagine that some paths will preserve the framework's shape, and some won't. In this research we explore the motion of an object having multiple forces applied to it, therefore it is key to understand what happens to a framework when velocities are assigned to the nodes. To that end, instead of looking at two frameworks at the same time, we could compare the same framework at two different times. This idea will enable us to characterize a framework in motion.

A framework is equivalent over time if

$$\tfrac{d}{dt}\|d_k\left(t\right)\|^2 = 0 \quad k = 1, 2, \cdots, \varepsilon. \tag{2.27}$$

In other words, a framework is equivalent over time if the length of the edges remains unchanged as the framework moves along its predetermined trajectory. Now, if the distance between *any* two nodes (whether there is an edge connecting them or not) in the framework is constant over time, we say that the framework is congruent over time. Formally, a framework is equivalent over time if

$$\|p_{ij}(t_1)\| = \|p_{ij}(t_2)\| \quad (i, j) \in \mathcal{E} \quad \forall t_1, t_2 \ . \tag{2.28}$$

where t_1 and t_2 are any (and every) two times at which the edges lengths are measured. Similarly, a framework is congruent over time if

$$\|p_{ij}(t_1)\| = \|p_{ij}(t_2)\| \quad i, j \in \mathcal{V} \quad \forall t_1, t_2 \ . \tag{2.29}$$

Note that (2.27) has to be satisfied for all the edges in the graph. Take for instance the example of Figure 2.4:

$$\frac{d}{dt}\|p_{ij}\|^2 = \frac{d}{dt}\left(p_{ij}^T p_{ij}\right) = 2p_{ij}^T \dot{p}_{ij} \tag{2.30}$$

$$\begin{aligned}
d_1 = p_{12} &\Rightarrow p_{12}^T \dot{p}_{12} = 0 \\
d_2 = p_{23} &\Rightarrow p_{23}^T \dot{p}_{23} = 0 \\
d_3 = p_{24} &\Rightarrow p_{24}^T \dot{p}_{24} = 0 \\
d_4 = p_{34} &\Rightarrow p_{34}^T \dot{p}_{34} = 0.
\end{aligned} \tag{2.31}$$

These 4 equations can be written in matrix form, as they describe a system of linear equations,

$$\begin{bmatrix} p_{12}^T & -p_{12}^T & 00 & 00 \\ 00 & p_{23}^T & -p_{23}^T & 00 \\ 00 & p_{24}^T & 00 & -p_{24}^T \\ 00 & 00 & p_{34}^T & -p_{34}^T \end{bmatrix} \begin{bmatrix} \dot{p}_1 \\ \dot{p}_2 \\ \dot{p}_3 \\ \dot{p}_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{2.32}$$

The matrix in (2.32) is called the *rigidity matrix* of the framework, denoted $R(p)$, and the vector it's multiplying is called an *infinitesimal motion.* In general, (2.27) leads to ε equations that have to be satisfied, which in turn lead to the matrix form:

$$R(p)u = 0_{\varepsilon\times 1}, \tag{2.33}$$

where $R(p) \in \mathbb{R}^{\varepsilon\times 2n}$, and $u \in \mathbb{R}^{2n}$ is the infinitesimal motion (a set of velocities that when applied to the nodes, equivalency is preserved). Observe that for any framework, the rigid body translations and rotations will always be infinitesimal motions - these are referred to as the *trivial motions* of the framework, and can be expressed as

$$\begin{aligned} u_1 &= \mathbb{1}_n \otimes \begin{bmatrix} 1 & 0 \end{bmatrix}^T \\ u_2 &= \mathbb{1}_n \otimes \begin{bmatrix} 0 & 1 \end{bmatrix}^T \\ u_3(p) &= (I_n \otimes T)\, p, \end{aligned} \tag{2.34}$$

where p is the concatenation of the position vectors $p = \begin{bmatrix} p_1^T & \cdots & p_n^T \end{bmatrix}^T$ and T is the rotation matrix defined in (2.2).

This intuitive definition of the rigidity matrix is useful because it gives us an insight on what it represents; however to formally define it we're going to make use of the *edge function* of the framework

$$f_{\mathcal{G}}(p) : \mathbb{R}^{2n} \to \mathbb{R}^{\varepsilon}. \tag{2.35}$$

This function receives the concatenation of all the positions of the nodes in the framework, and returns a concatenation of (half) the lengths of the edges (squared):

$$f_{\mathcal{G}}(p) = \frac{1}{2}\begin{bmatrix} \|d_1\|^2 & \cdots & \|d_\varepsilon\|^2 \end{bmatrix}^T. \tag{2.36}$$

In the example of Figure 2.4,

$$f_{\mathcal{G}}(p) = \frac{1}{2}\begin{bmatrix} 10 & 10 & 10 & 2 \end{bmatrix}^T. \tag{2.37}$$

With the edge function in hand, the rigidity matrix is simply the *Jacobian* of the edge function,

$$R(p) = \frac{\partial f_{\mathcal{G}}(p)}{\partial p}. \tag{2.38}$$

We can see through this definition that the rigidity matrix is the first term in the *Taylor* series of the edge function. Therefore motions in the null space of $R(p)$ (or infinitesimal motions) maintain the lengths of the edges to first order. This rigidity matrix will be of help when trying to find just "how rigid" a framework is, but in order to do that, we first provide some additional formal definitions of rigidity.

Definition 2.3.3 (Rigid frameworks)**.** A framework $\mathcal{F}_1(\mathcal{G}, P_1)$ is *rigid* if there exists an $\epsilon > 0$ such that every framework $\mathcal{F}_2(\mathcal{G}, P_2)$ that is equivalent to $\mathcal{F}_1(\mathcal{G}, P_1)$ and satisfies $||p_1(v) - p_2(v)|| < \epsilon \ \forall v \in \mathcal{V}$, is congruent to $\mathcal{F}_1(\mathcal{G}, P_1)$.

Definition 2.3.4 (Globally rigid frameworks)**.** A framework is *globally rigid* if every other framework that is equivalent to it, is also congruent.

This definition is fairly straightforward, we say that a framework is globally rigid when equivalency and congruency lead to the same mathematical condition. Finally, we can discuss the type of rigidity that is most relevant to this work, that is, *infinitesimal rigidity*. Recall the intuitive definition of the rigidity matrix, explained through the steps of (2.27) - (2.32); we introduced the rigidity matrix and the infinitesimal motions.

Definition 2.3.5 (Infinitesimally rigid frameworks)**.** A framework is *infinitesimally rigid* if the only infinitesimal motions that it has are rotations and translations.

The connection between the infinitesimal motions and the rigidity matrix can be seen in the following lemma

Lemma 2.3.6 ([14])**.** *A framework in $\mathbb{R}^2$ is infinitesimally rigid if and only if*

$$\text{rank}\{R(p)\} = 2n - 3.$$

In the case of Lemma 2.3.6 we can see that to span the null space of $R(p)$ we'll need 3 vectors. These 3 vectors are the translations and rotations mentioned above. Take for instance u_1: it is composed of n times the vector $\begin{bmatrix} 1 & 0 \end{bmatrix}^T$, and the framework has also n nodes; so if we look at each vector $\begin{bmatrix} 1 & 0 \end{bmatrix}^T$ as a velocity to be applied at the nodes we would indeed end up translating the framework in the $\hat{x}$ direction. In the same manner u_2 would cause a translation in the $\hat{y}$ direction, and $u_3(p)$ would cause a rotation around the centroid of the formation.

An interesting interpretation of these null space vectors is described in [16]. There, instead of computing the null space of $R(p)$, the authors compute the eigenvectors of the matrix $R(p)^T R(p)$, termed the *symmetric rigidity matrix*. This matrix has $2n$ eigenvalues and corresponding eigenvectors, ordered as follows:

$$0 = \lambda_1 = \lambda_2 = \lambda_3 \leq \cdots \leq \lambda_{2n}. \tag{2.39}$$

The matrix $R(p)^T R(p)$ is symmetric and positive semi-definite, therefore its eigenvalues are real and non-negative. In addition, if the framework is infinitesimally rigid, and in

$\mathbb{R}^2$, $R(p)^T R(p)$ will have exactly 3 zero eigenvalues. Thus, for an infinitesimally rigid framework in the plane, it follows that $\lambda_1 = \lambda_2 = \lambda_3 = 0$ and $\lambda_i > 0$ for $i = 4, \ldots, 2n$.

This means that the eigenvectors corresponding to the first 3 eigenvalues span the null space of the matrix, and are no other than $u_1, u_2, u_3(p)$ depicted in (2.34) (the rigidity matrix and its symmetric counterpart share the same null space). In other words the first 3 eigenvectors cause trivial motions on the framework. This poses the following question: what do the *rest* of the eigenvectors cause? To answer this question let us take a closer look at the eigenvectors of $R(p)^T R(p)$,

$$R(p)^T R(p)\, u^\ell = \lambda_\ell u^\ell. \tag{2.40}$$

Here, u^ℓ is the eigenvector and λ_ℓ is the eigenvalue; $\ell = 1, 2, \cdots, 2n$. Now let us define

$$R(p)\, u^\ell = w^\ell \in \mathbb{R}^\varepsilon \tag{2.41}$$

$$w^\ell = \begin{bmatrix} w_1^\ell & \cdots & w_k^\ell & \cdots & w_\varepsilon^\ell \end{bmatrix}^T, \tag{2.42}$$

such that $w_k^\ell \equiv w_{ij}^\ell$ since edge k connects nodes i and j, and

$$R(p)^T w^\ell = \lambda_\ell u^\ell. \tag{2.43}$$

Now (2.40) can be rewritten as

$$\sum_{j \in \mathcal{N}_i} w_{ij}^\ell p_{ij} = \lambda_i u_i^\ell. \tag{2.44}$$

Note that $\mathcal{N}_i$ marks the neighborhood of node i. Also, the eigenvector u^ℓ has $2n$ entries, or n *pairs* of entries,

$$u^\ell = \begin{bmatrix} u_1^\ell & \cdots & u_i^\ell & \cdots & u_n^\ell \end{bmatrix}^T \tag{2.45}$$

$$u_i^\ell = \begin{bmatrix} u_{i,x}^\ell & u_{i,y}^\ell \end{bmatrix}^T. \tag{2.46}$$

This representation is useful because it will help answer the question of what will the rest of the eigenvectors (not u_1, u_2 and $u_3(p)$) cause on the framework when applied as forces on the vertices. We can view w_{ij}^ℓ as the magnitude of the force (per unit length) exerted on the edge connecting nodes i and j, through the nodes i and j. When $w_{ij}^\ell > 0$ the force is called a *tension*, and when $w_{ij}^\ell < 0$ the force is called a *compression*. This visualization of forces on edges leads us to finding the forces that, along with the tensions and compressions, reach an equilibrium:

$$f_i + \sum_{j \in \mathcal{N}_i} w_{ij}^\ell p_{ij} = \begin{bmatrix} 0 & 0 \end{bmatrix}^T. \tag{2.47}$$

If there exist scalars w_{ij}^{ℓ} satisfying equation 2.47, we say that f_i is a *resolvable force*. But since $\sum_{j \in \mathcal{N}_i} w_{ij}^{\ell} p_{ij} = \lambda_i u_i^{\ell}$, we can rewrite (2.47):

$$f_i = -\lambda_i u_i^{\ell}, \tag{2.48}$$

which leads to Theorem 2.1, by [16]:

Theorem 2.1 ([16]). *Let $\mathcal{F}$ be an infinitesimally rigid framework in $\mathbb{R}^2$. Every vector $f_i = -\lambda_i u_i^{\ell}$ for any $i = 4, 5, \cdots, 2n$ is a resolvable force, where $u_i,\ i = 1, 2, 3$ are the eigenvectors of the symmetric rigidity matrix corresponding to the zero eigenvalue.*

The vectors that span the null space of $R(p)$ cause pure motions to the framework when applied as velocities. In this work we show the effect of the null space and image of $R(p)$ when applied as forces to a rigid body attached to a formation of robots, including the case where the COM of the system (robots plus object) does not coincide with the centroid of the framework.

Chapter 3

Dynamics of Cooperative Object Manipulation

The first step towards moving an object cooperatively, should be to define how it will be done in terms of the setup. In this work we consider the manipulation of a square table placed on top of a group of mobile robots. We assume the robots are mechanically fixed to the table such that the robots can rotate with respect to the table, but cannot shift.

In this chapter we model the dynamics of a system composed of an object along with n robots whose total mass and moment of inertia about the COM are m and I respectively. In this setup the robots are the intermediary of the floor and the object to be moved, and therefore the robots are subject to friction, but not the payload. The center of mass (COM) of the system (that is, the object and the robots together) is marked by the vector p_S with respect to a stationary FOC, as depicted in Figure 2.2. Here, p_i is the position of robot i with respect to the stationary FOC, and r_i is the position of robot i with respect to the COM, $r_i = p_i - p_S$. The angle θ marks the orientation of the object, f_i is the force applied by robot i and F_i is the friction the robot is subject to, modeled here as $F_i = -\frac{\mu mg}{n}\dot{p}_i$, where μ is the friction coefficient between the robot and the floor, and g is the gravitational constant. With this setup the equations of motion are as derived in (2.13),

$$\begin{aligned} m\ddot{p}_S &= \left(\mathbb{1}_n^T \otimes I_2\right) f - \mu mg \left(\dot{p}_S + \dot{\theta} T \bar{r}\right) \\ \mathrm{I}\ddot{\theta} &= f^T \left(I_n \otimes T\right) r - \mu mg \left(\dot{p}_S^T T \bar{r} + \dot{\theta}\frac{r^T r}{n}\right). \end{aligned} \tag{3.1}$$

3.1 State Space Definition

In this part we define a state space to be used, and we implement it in the equations of motion (2.13),

$$\begin{aligned} \chi_1 &= \theta \\ \chi_2 &= \dot{\theta} \\ \chi_3 &= p_S \\ \chi_4 &= \dot{p}_S, \end{aligned} \tag{3.2}$$

this leads to the following state-space representation of the dynamics

$$\begin{aligned} \dot{\chi}_1 &= \chi_2 \\ \dot{\chi}_2 &= \frac{1}{\mathrm{I}}\left[f^T\left(I_n \otimes T\right) r - \mu m g\left(\chi_4^T T \bar{r} + \chi_2 \frac{r^T r}{n}\right)\right] \\ \dot{\chi}_3 &= \chi_4 \\ \dot{\chi}_4 &= \frac{1}{m}\left[\left(\mathbb{1}_n^T \otimes I_2\right) f - \mu m g\left(\chi_4 + \chi_2 T \bar{r}\right)\right]. \end{aligned} \tag{3.3}$$

The first part in the analysis of these non-linear dynamics is to check the effect of simple forces in the symmetric case. If we were to grab a square table by the corners and pull to the right (for instance), we would expect to see the table moving to the right. Next we corroborate our intuition with numerical simulations.

3.2 Symmetric Analysis

In this section we assume that the robots are positioned symmetrically around the table's center of mass, and then apply u_1, u_2 and $u_3(p)$ as forces. A symmetric positioning (shown in Figure 3.1) means that the sum of the positions of the robots is zero both in the $\hat{x}$ and $\hat{y}$ coordinates, with respect to the center of mass of the system. In other words,

$$\bar{r} = \begin{bmatrix} 0 & 0 \end{bmatrix}^T. \tag{3.4}$$

When substituting (3.4) in the dynamics, we arrive at the following simplified version,

$$\begin{aligned} \dot{\chi}_1 &= \chi_2 \\ \dot{\chi}_2 &= \frac{1}{\mathrm{I}}\left[f^T\left(I_n \otimes T\right) r - \mu m g \chi_2 \frac{r^T r}{n}\right] \\ \dot{\chi}_3 &= \chi_4 \\ \dot{\chi}_4 &= \frac{1}{m}\left[\left(\mathbb{1}_n^T \otimes I_2\right) f - \mu m g \chi_4\right]. \end{aligned} \tag{3.5}$$

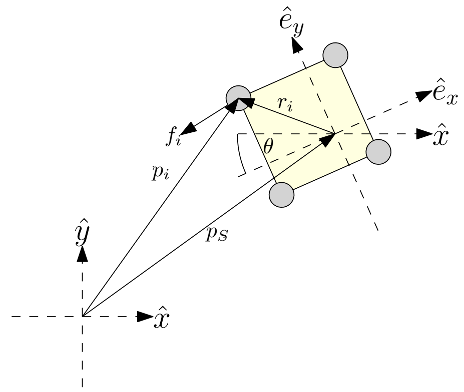


Figure 3.1: A rigid body moved by n robots positioned at the corners.

To start, we apply u_1 (defined in 2.34) as a force, and examine the resulting trajectory of the system,

$$f = \mathbb{1}_n \otimes \begin{bmatrix} 1 & 0 \end{bmatrix}^T. \tag{3.6}$$

This force represents the case where all robots pull the table to the right, and it appears explicitly only in the linear and angular acceleration, that is, $\dot{\chi}_2$ and $\dot{\chi}_4$,

$$\begin{aligned} \dot{\chi}_2 &= \frac{1}{\mathrm{I}} \left[\left(\mathbb{1}_n \otimes \begin{bmatrix} 1 & 0 \end{bmatrix}^T \right)^T (I_n \otimes T)\, r - \mu m g \chi_2 \frac{r^T r}{n} \right] \\ &= -\mu m g \chi_2 \frac{r^T r}{\mathrm{I} n} \\ \dot{\chi}_4 &= \frac{1}{m} \left[\left(\mathbb{1}_n^T \otimes I_2 \right) \left(\mathbb{1}_n \otimes \begin{bmatrix} 1 & 0 \end{bmatrix}^T \right) - \mu m g \chi_4 \right] \\ &= -\mu g \chi_4 + \begin{bmatrix} n/m & 0 \end{bmatrix}^T. \end{aligned} \tag{3.7}$$

Note that $r^T r$ is constant over time:

Proposition 3.2.1. *For a system composed of n robots rigidly attached to an object, the size $r^T r$ is constant over time. In other words:*

$$\frac{d\left(r^T r\right)}{dt} = 0. \tag{3.8}$$

Proof.

$$r^T r = \sum_{i=1}^{n} \underbrace{\|r_i\|}_{const}^2, \tag{3.9}$$

because even though r_i changes over time, its size does not (the robots are modeled as "bolted" to the object and therefore their distance to the object's COM is constant).□

Hence these now decoupled equations turn into a linear system that can be solved explicitly,

$$\begin{aligned} \chi_2(t) &= e^{-\mu m g \frac{r^T r}{\mathrm{I} n} t} \chi_2(0) \\ \chi_4(t) &= \begin{bmatrix} \frac{n/m}{\mu g} \\ 0 \end{bmatrix} \left(1 - e^{-\mu g t}\right) + \chi_4(0) e^{-\mu g t} \end{aligned} \tag{3.10}$$

The meaning of (3.10), is that even if the system starts with some angular velocity $\chi_2(0)$, that velocity will decay and only a constant linear velocity will remain. This corroborates our intuition, and furthermore, it can be seen in the following simulation, whose parameters are as follows: the mass of the system is $m = 32.7\,[kg]$, the friction coefficient is $\mu = 0.25$, the system's moment of inertia is $\mathrm{I} = 52.0447\,\left[kgm^2\right]$, the number of robots is $n = 4$, the gravitational constant is $g = 9.81\,\left[\frac{m}{s^2}\right]$, the force applied by each robot is $f_i = \begin{bmatrix} 5 & 0 \end{bmatrix}^T [N]$, the initial position of the robots is at the corners of the table, the initial position of the COM is at the origin - $\chi_3(0) = \begin{bmatrix} 0 & 0 \end{bmatrix}^T [m]$, the initial linear velocity of the COM is $\chi_4(0) = \begin{bmatrix} 0 & 0 \end{bmatrix}^T \left[\frac{m}{s}\right]$, the initial angular velocity is $\chi_2(0) = 0.5\,\left[\frac{rad}{s}\right]$, and the initial orientation is $\chi_1(0) = 30\,[deg]$.

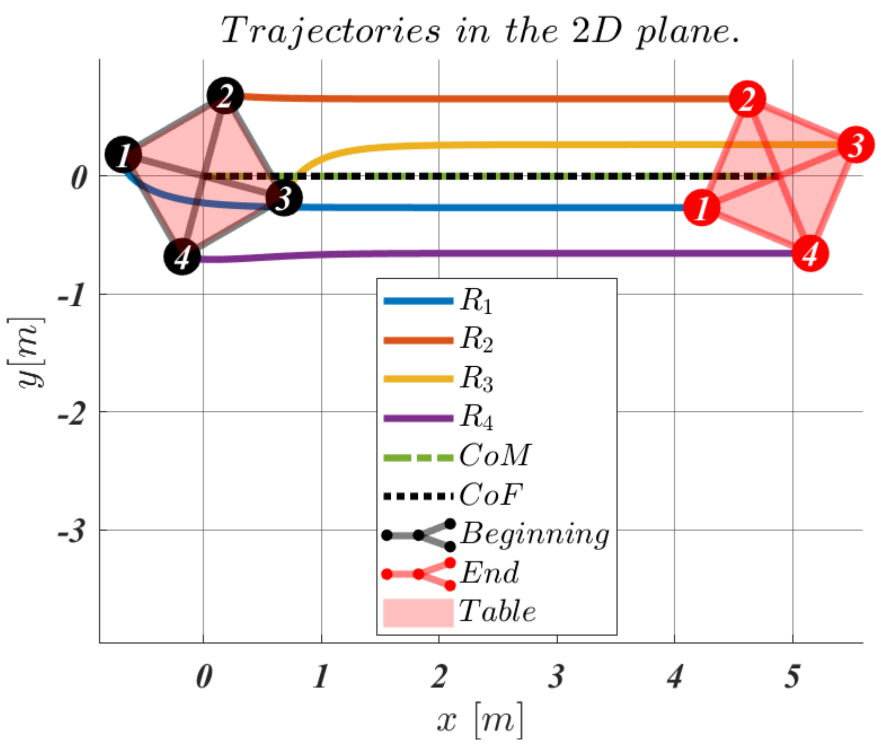
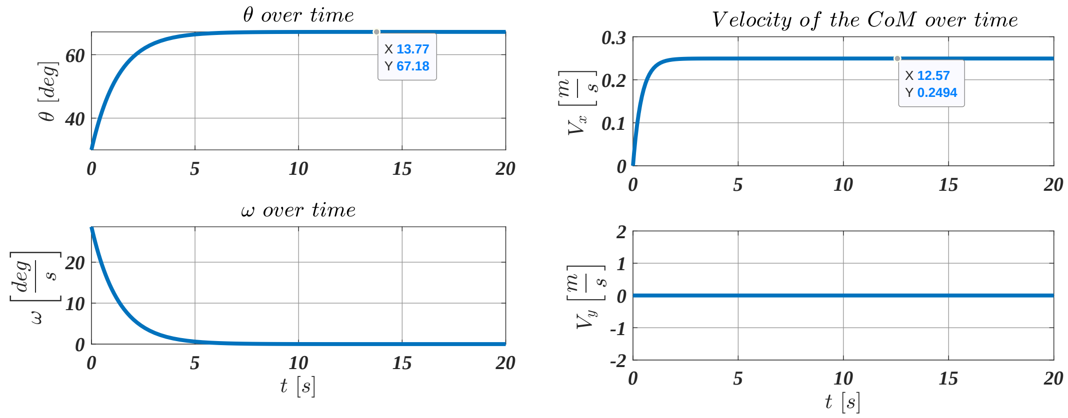


Figure 3.2: The planar motion caused by pulling to the right from the corners of the table. *CoM* marks the center of mass of the system, which in this symmetric case coincides with the centroid of the formation *CoF*.



(a) The orientation and angular velocity of the table over time.

(b) The linear velocity of the table over time.

Figure 3.3: After applying the force u_1, the table develops a steady state translation in spite of the inital rotational velocity.

Figure 3.2 shows the resulting trajectory of the table and robots, and it can be seen that the angular motion decays as expected. This can also be seen in the Figure 3.3a. Naturally the analysis of the force $f = \mathbb{1}_n \otimes \begin{bmatrix} 0 & 1 \end{bmatrix}^T$ is similar to the one just presented, and therefore will be omitted. However the effect of the rotating force $f = (I_n \otimes T)\, p$ is not as straightforward.
The equations of motion for this case are

$$\begin{aligned}
\dot{\chi}_2 &= \frac{1}{\mathrm{I}} \left\{ \left[(I_n \otimes T)\, p\right]^T (I_n \otimes T)\, r - \mu m g \chi_2 \frac{r^T r}{n} \right\} \\
&= \frac{r^T r}{\mathrm{I}} \left\{ 1 - \chi_2 \frac{\mu m g}{n} \right\} \\
\dot{\chi}_4 &= \frac{1}{m} \left\{ \left(\mathbb{1}_n^T \otimes I_2\right) \left[(I_n \otimes T)\, p\right] - \mu m g \chi_4 \right\} \\
&= \frac{1}{m} \left\{ n T \chi_3 - \mu m g \chi_4 \right\}
\end{aligned} \tag{3.11}$$

So we can rewrite our system as

$$\dot{\chi} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \;\; 0 & 0 \;\; 0 \\ 0 & -\frac{\mu m g r^T r}{n I} & 0 \;\; 0 & 0 \;\; 0 \\ 0 & 0 & 0 \;\; 0 & 1 \;\; 0 \\ 0 & 0 & 0 \;\; 0 & 0 \;\; 1 \\ 0 & 0 & \frac{n}{m} T & -\mu g I_2 \\ 0 & 0 & & \end{bmatrix}}_{A} \chi + \underbrace{\begin{bmatrix} 0 \\ \frac{r^T r}{I} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{B} . \tag{3.12}$$

From (3.12) we can see that the dynamics can be decoupled. The angular part of the dynamics can be rewritten as two scalar equations and be solved immediately:

$$\chi_1(t) = \frac{n\mathrm{I}}{\mu mgr^T r}\left[\chi_2(0) - \frac{n}{\mu mg}\right]\left(1 - e^{-\frac{\mu mgr^T r}{n\mathrm{I}}t}\right) + \frac{nt}{\mu mg} + \chi_1(0) \tag{3.13}$$

$$\chi_2(t) = \left[\chi_2(0) - \frac{n}{\mu mg}\right] e^{-\frac{\mu mgr^T r}{n\mathrm{I}}t} + \frac{n}{\mu mg}. \tag{3.14}$$

The linear part can be rewritten as:

$$\begin{bmatrix} \dot{\chi}_3 \\ \dot{\chi}_4 \end{bmatrix} = \underbrace{\begin{bmatrix} 0_2 & I_2 \\ \frac{n}{m}T & -\mu g I_2 \end{bmatrix}}_{A} \begin{bmatrix} \chi_3 \\ \chi_4 \end{bmatrix}, \tag{3.15}$$

where 0_2 is a 2×2 zeros matrix. The solution to this system is then:

$$\begin{bmatrix} \chi_3(t) \\ \chi_4(t) \end{bmatrix} = e^{At} \begin{bmatrix} \chi_3(0) \\ \chi_4(0) \end{bmatrix}. \tag{3.16}$$

The solution presented in (3.16) shows that the initial conditions play an important role in this case, since the matrix A has eigenvalues in the open right half plane (this can be seen from the Routh-Hurwiz criterion, since the coefficient of λ^1 is zero):

$$\det[\lambda I_4 - A] = 0 \Leftrightarrow \tag{3.17}$$

$$\lambda^4 + 2\mu g\lambda^3 + \mu^2 g^2 \lambda^2 + (n/m)^2 = 0. \tag{3.18}$$

In other words for any non-zero initial conditions the linear part of the system (3.16) will grow unbounded, and this is also shown in simulation. Figure 3.4 shows the trajectory of the system from zero initial conditions (applying the force $u_3(p)$, where all the states start from zero) :

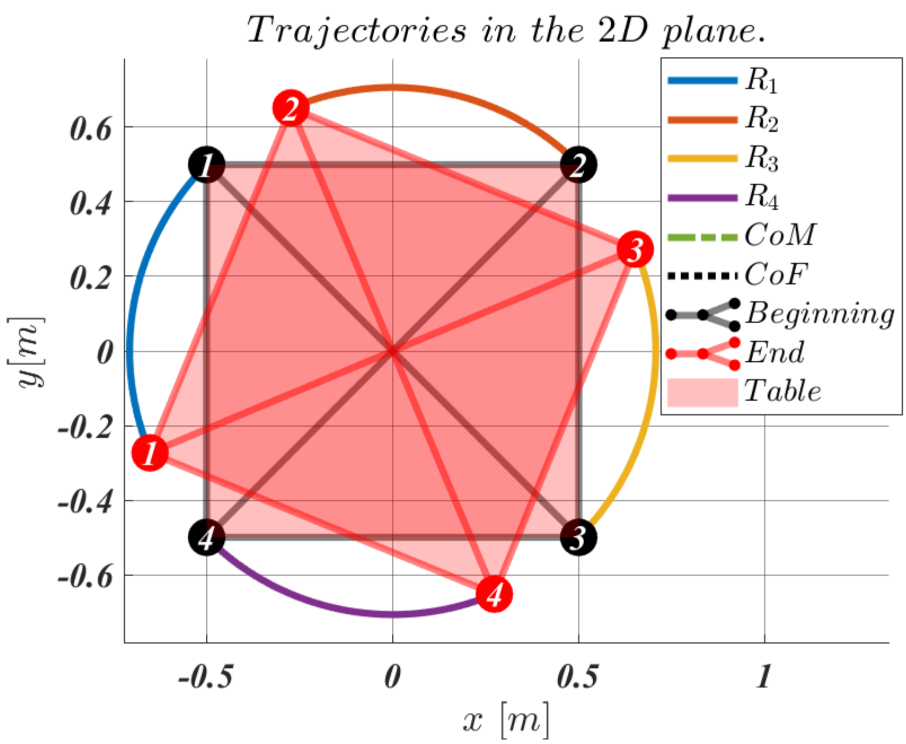
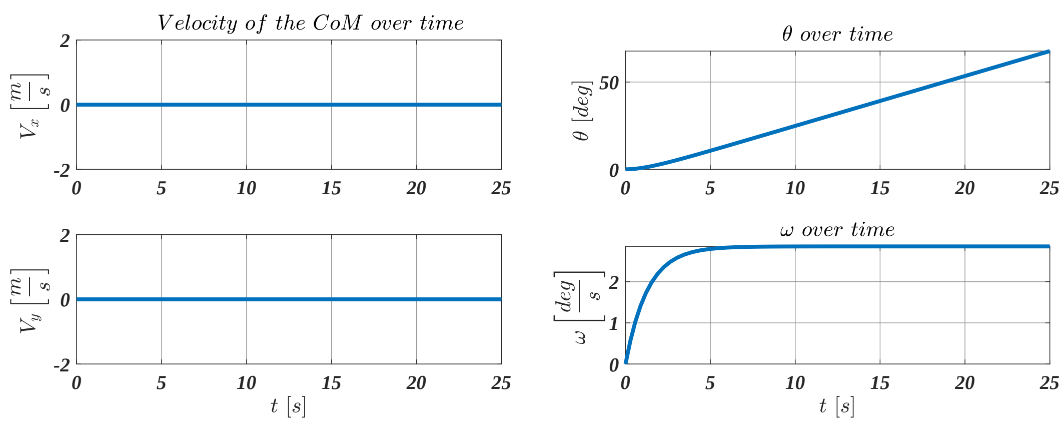


Figure 3.4: The planar motion caused by applying the force $f = (I_n \otimes T)\, p$ on the rigid body. In this symmetric case the result is a pure rotation about the centroid of the formation, and the center of mass of the system, which coincide.



(a) The linear velocity of the CoM remains zero as expected.

(b) The angular velocity and orientation of the table.

Figure 3.5: The linear and angular velocities of the table, from zero initial conditions.

From looking at these results it seems that the rotating force causes a pure rotation about the COM, however the moment we change the initial conditions the motion is not a pure rotation anymore. It is instead, a rotation and a translation. The following graphs show the result of the same simulation, with the only difference being in the initial linear velocity, randomly chosen to be: $\chi_4(0) = \begin{bmatrix} \pi & \sqrt{e} \end{bmatrix}^T$.

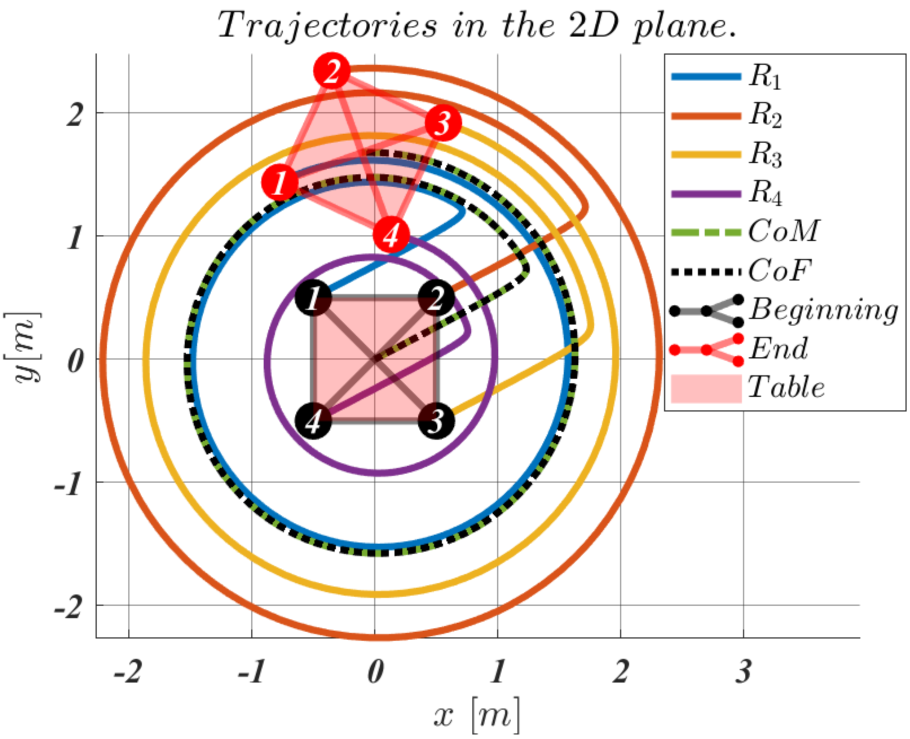
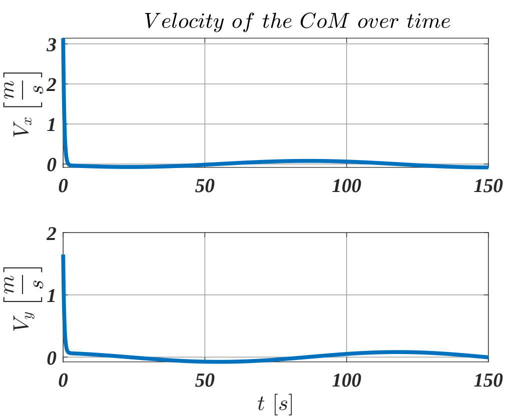
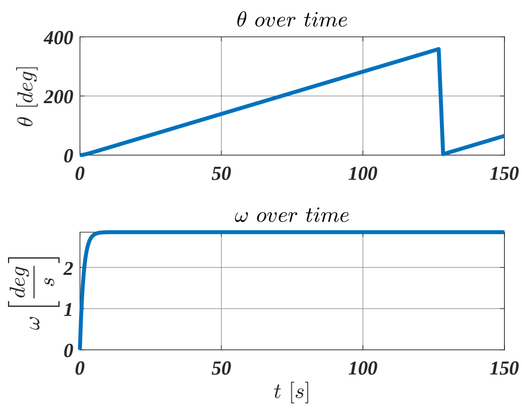


Figure 3.6: The planar motion caused by applying the force $u_3(p)$ on the table. Starting with an initial shift, the system does not rotate as intended.

These results show clearly that even though the first two trivial motions cause pure translations, the last trivial motion $u_3(p)$ caused a pure rotation only from zero initial conditions. The motivation of this work, is to compute forces that create pure motions when applied on an object, and this analysis shows us that to do so we need to come up with a more reliable input.



(a) The linear velocity of the CoM oscillates.



(b) The angular velocity and orientation of the table.

Figure 3.7: The linear and angular velocities of the table, with a linear velocity as initial conditions.

3.3 Asymmetric Analysis

In this section we no longer assume that the robots are positioned symmetrically around the table's center of mass. That is, they are randomly positioned around the table. We then apply u_1, u_2 and $u_3(p)$ as forces. The dynamics in the general positioning case are depicted in (3.3) and are too complex to solve. For this reason, to see the effect of the discussed forces on the system, a simulation was run and the results are presented in chapter 5.

For the case where $f = u_1$ or $f = u_2$, we can see in Figures 5.1 and 5.3 that when starting from rest, the forces u_1 and u_2 create translations in $\hat{x}$ and $\hat{y}$, respectively. In addition, we may see in Figures 5.2b and 5.4b that the translations are pure only after the transient dies out.

For the case where $f = u_3(p)$, we can see in section 5.2 that for all the initial conditions that were tried, the force creates a combination of translations and rotations. Meaning that the force does not create a pure rotation.

Chapter 4

Desired Motion

In this work we are interested in computing the forces required to enable an object to follow a desired path. Since we're dealing with a general case where the path is unknown, we should be able to produce all basic motions - that is, rotations and translations. In order to do that, we look at the steady state of the dynamics of the system, and translate the words "rotations and translations at a constant velocity" into mathematical conditions that these dynamics should satisfy. For instance, a steady state translation is a type of motion characterized by no accelerations (that's the steady state part), and a constant linear velocity in the desired direction ($\hat{x}$ or $\hat{y}$). Similarly, a steady state rotation is a type of motion characterized by no accelerations and a constant angular velocity.
Going back to enabling an object to go from a to b, we shouldn't forget about a very important part - which is to stop after we reach b. In other words, another force that should be found within this context, is a force that causes no motion. This force can be seen as an equilibrium force, since the *motion* that it produces is characterized by no accelerations nor velocities.

As mentioned above, a steady state trajectory is characterized by no acceleration, that is:

$$\dot{\chi}_2 = 0 \tag{4.1}$$

$$\dot{\chi}_4 = \begin{bmatrix} 0 & 0 \end{bmatrix}^T \tag{4.2}$$

when substituting this into the dynamics of the system (3.3), we arrive at:

$$\dot{\chi}_2 = 0 = \frac{1}{\mathrm{I}}\left[f^T\left(I_n \otimes T\right)r - \mu mg\left(\chi_4^T T\bar{r} + \chi_2\frac{r^T r}{n}\right)\right] \tag{4.3}$$

$$\Rightarrow f^T\left(I_n \otimes T\right)r = \mu mg\left((\chi_4^{ss})^T T\bar{r} + \chi_2^{ss}\frac{r^T r}{n}\right) \tag{4.4}$$

$$\dot{\chi}_4 = 0 = \frac{1}{m}\left[\left(\mathbb{1}_n^T \otimes I_2\right)f - \mu mg\left(\chi_4 + \chi_2 T\bar{r}\right)\right] \tag{4.5}$$

$$\Rightarrow \left(\mathbb{1}_n^T \otimes I_2\right)f = \mu mg\left(\chi_4^{ss} + \chi_2^{ss}T\bar{r}\right), \tag{4.6}$$

where $\bar{r}$ is defined in (2.1). Alternatively we can rewrite these equations in matrix form:

$$\underbrace{\begin{bmatrix} r^T\left(I_n \otimes T\right) \\ \left(\mathbb{1}_n^T \otimes I_2\right) \end{bmatrix}}_{M} f = \underbrace{\mu mg\begin{bmatrix} -(\chi_4^{ss})^T T\bar{r} - \chi_2^{ss}\frac{r^T r}{n} \\ \chi_4^{ss} + \chi_2^{ss}T\bar{r} \end{bmatrix}}_{B}. \tag{4.7}$$

The matrix $M \in \mathbb{R}^{3\times 2n}$ is the *motion* matrix of the system, and the vector B dictates the ss motion that the force f will create. Note that the matrix M is always of full rank, since this is a representation of a physical system, and two robots cannot occupy the same spot.

Proposition 4.0.1. *The motion matrix M is of full rank, i.e.,* $\mathrm{rank}\{M\} = 3$.

Proof. If $\mathrm{rank}\{M\} \neq 3$ then it is either 2 or 1. The proof that $\mathrm{rank}\{M\} \neq 1$ is trivial and therefore omitted. Assume by contradiction that $\mathrm{rank}\{M\} = 2$. Without loss of generality, the first row of M can be written as a linear combination of the last 2 rows, i.e., there exists some constants c_1 and c_2 such that

$$\begin{bmatrix} 1 & c_1 & c_2 \end{bmatrix} M = \begin{bmatrix} 0 & \cdots & 0 \end{bmatrix}.$$

This implies that $r_i = \begin{bmatrix} -c_1 & -c_2 \end{bmatrix}^T$ for $i = 1, \ldots, n$ which is not a valid positioning (two robots cannot occupy the same spot), leading to a contradiction. □

In this chapter we discuss the steady state (ss) trajectories we'd like the body to follow, and find the forces to be applied in order for that to happen. In general, any in-plane motion can be described as a sum of translations and rotations; therefore, if the aim is to be able to move an object from a to b, we should be able to produce these basic motions.

4.1 Equilibrium

In this section we look for the forces that will bring the system to a steady state equilibrium, that is - no motion whatsoever,

$$\begin{aligned} \dot{\chi}_1 &= 0 \\ \dot{\chi}_2 &= 0 \\ \dot{\chi}_3 &= \begin{bmatrix} 0 & 0 \end{bmatrix}^T \\ \dot{\chi}_4 &= \begin{bmatrix} 0 & 0 \end{bmatrix}^T. \end{aligned} \tag{4.8}$$

When substituting this constraints and solving for f:

$$\begin{bmatrix} r^T \left(I_n \otimes T\right) \\ \left(\mathbb{1}_n^T \otimes I_2\right) \end{bmatrix} f = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{4.9}$$

In other words, we need to find the null space of M. Since the matrix is of rank 3, and has $2n$ columns, we'll need $2n-3$ null space vectors, marked as $f^{\mathcal{N}_i}$, where

$$f^{\mathcal{N}_i} = \begin{bmatrix} f_1^{\mathcal{N}_i} & \cdots & f_k^{\mathcal{N}_i} & \cdots & f_{2n}^{\mathcal{N}_i} \end{bmatrix}^T. \tag{4.10}$$

In this representation, entry number k in the null space vector is defined as

- If i is odd

$$\begin{aligned} f_k^{N_i} &= \begin{cases} -\frac{r_{j1}^x}{r_{21}^y} & k=1 \\ -1 & k=2 \\ \frac{r_{j1}^x}{r_{21}^y} & k=3 \\ 1 & k=i+3 \\ 0 & otherwise \end{cases} \\ j &= \frac{i+3}{2} \end{aligned} \tag{4.11}$$

- If i is even

$$\begin{aligned} f_k^{N_i} &= \begin{cases} \frac{r_{j2}^y}{r_{21}^y} & k=1 \\ -\frac{r_{j1}^y}{r_{21}^y} & k=3 \\ 1 & k=i+3 \\ 0 & otherwise \end{cases} \\ j &= \frac{i+4}{2} \end{aligned} \tag{4.12}$$

Proposition 4.1.1. *In the general case, where the positioning of the robots is not necessarily symmetric about the center of mass, an equilibrium can be achieved in steady* *state, by applying the forces described in* (4.10) *through the robots, regardless of the initial conditions.*

Proof. To prove this claim, we need to substitute the proposed forces into the steady state dynamics and show that the resulting vector corresponds to the desired motion, depicted in (4.8).

Let us start with the case where i is odd. In detail, the motion matrix is:

$$M = \begin{bmatrix} r_1^y & -r_1^x & r_2^y & -r_2^x & \cdots & r_n^y & -r_n^x \\ 1 & 0 & 1 & 0 & \cdots & 1 & 0 \\ 0 & 1 & 0 & 1 & \cdots & 0 & 1 \end{bmatrix}. \tag{4.13}$$

The nullspace vector $f^{\mathcal{N}_i}$, in this case is:

$$f^{\mathcal{N}_i} = \begin{bmatrix} -\frac{r_{j1}^x}{r_{21}^y} & -1 & \frac{r_{j1}^x}{r_{21}^y} & \underbrace{\cdots}_{*} \end{bmatrix}^T, \tag{4.14}$$

note that there's a 1 somewhere in $*$ (precisely in entry number $i+3$), and all the rest of the entries are zeros. To prove that $f^{\mathcal{N}_i}$ is in the null space of M, we need to show that $Mf^{\mathcal{N}_i} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$, and indeed

$$Mf^{\mathcal{N}_i} = \begin{bmatrix} -\frac{r_{j1}^x}{r_{21}^y} r_1^y + r_1^x + \frac{r_{j1}^x}{r_{21}^y} r_2^y - r_j^x \\ -\frac{r_{j1}^x}{r_{21}^y} + \frac{r_{j1}^x}{r_{21}^y} \\ -1+1 \end{bmatrix}. \tag{4.15}$$

In the case where i is even, the nullspace vector $f^{\mathcal{N}_i}$ is:

$$f^{\mathcal{N}_i} = \begin{bmatrix} \frac{r_{j2}^y}{r_{21}^y} & 0 & -\frac{r_{j1}^y}{r_{21}^y} & \underbrace{\cdots}_{*} \end{bmatrix}^T \tag{4.16}$$

Again, there's a 1 in $*$, precisely in entry $i+3$ (note that now i is even, so $i+3$ is odd). Also in this case:

$$Mf^{\mathcal{N}_i} = \begin{bmatrix} \frac{r_{j2}^y}{r_{21}^y} r_1^y - \frac{r_{j1}^y}{r_{21}^y} r_2^y + r_j^y \\ \frac{r_{j2}^y}{r_{21}^y} - \frac{r_{j1}^y}{r_{21}^y} + 1 \\ 0 \end{bmatrix} \tag{4.17}$$

Since in both the odd and even cases $Mf^{\mathcal{N}_i} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$, the proposed forces $f^{\mathcal{N}_i}$ cause indeed a steady state equilibrium. □

4.2 Pure ss Rotation

In terms of the dynamics of the system, a pure ss rotation is defined by a constant angular velocity, and no linear motion at all. That is,

$$\begin{aligned} \dot{\chi}_1 &= const = \chi_2^{ss} \\ \dot{\chi}_2 &= 0 \\ \dot{\chi}_3 &= \begin{bmatrix} 0 & 0 \end{bmatrix}^T \\ \dot{\chi}_4 &= \begin{bmatrix} 0 & 0 \end{bmatrix}^T. \end{aligned} \tag{4.18}$$

The conditions of (4.18) pose a constraint on the dynamics, which can be used to find the force that will produce the desired motion. In order to do that, we need to substitute (4.18) in (4.7) and solve for f.

$$\begin{bmatrix} r^T (I_n \otimes T) \\ (\mathbb{1}_n^T \otimes I_2) \end{bmatrix} f = \mu m g \chi_2^{ss} \begin{bmatrix} -\frac{r^T r}{n} \\ T\bar{r} \end{bmatrix}. \tag{4.19}$$

Since M is of *full row rank* and for $n \geq 2$ we have $2n > 3$, it follows that $null\{M\} \neq \emptyset$, that is, it has a non-trivial nullspace. Therefore, the set of vectors solving $Mf = B$ includes some particular solution plus any vector in the null-space of M. The homogeneous solution is the set of all the vectors $f^{\mathcal{N}_i}$ found in the null space of M. The particular solution is simply:

$$f = f_{rot}^z = \frac{\mu m g}{n} \chi_2^{ss} (I_n \otimes T) r. \tag{4.20}$$

Proposition 4.2.1. *In the general case, where the positioning of the robots is not necessarily symmetric about the center of mass, a pure rotation can be achieved in steady state, by applying the force described in* (4.20) *through the robots, regardless of the initial conditions.*

Proof. To prove this claim, we need to substitute the proposed force into the steady state dynamics and show that the resulting vector corresponds to the desired motion, depicted in (4.18).

$$M f_{rot}^z = \begin{bmatrix} r^T (I_n \otimes T) \\ (\mathbb{1}_n^T \otimes I_2) \end{bmatrix} \frac{\mu m g}{n} \chi_2^{ss} (I_n \otimes T) r \tag{4.21}$$

$$= \mu m g \chi_2^{ss} \begin{bmatrix} \frac{r^T r}{n} \\ T\bar{r} \end{bmatrix}. \tag{4.22}$$

Since $M f_{rot}^z = \mu m g \chi_2^{ss} \begin{bmatrix} \frac{r^T r}{n} \\ T\bar{r} \end{bmatrix}$, the proposed force causes indeed a steady state rotation. □

4.3 Pure ss Translation

Similarly to the ss rotation, we need to define what a steady-state translation means in terms of the dynamics of the system, and then impose those constraints on the equations containing f. A ss translation means no angular movement, with a constant linear velocity,

$$\begin{aligned} \dot{\chi}_1 &= 0 \\ \dot{\chi}_2 &= 0 \\ \dot{\chi}_3 &= const = \chi_4^{ss} \\ \dot{\chi}_4 &= \begin{bmatrix} 0 & 0 \end{bmatrix}^T. \end{aligned} \tag{4.23}$$

This, in turn, leads to the following:

$$\begin{bmatrix} r^T (I_n \otimes T) \\ \mathbb{1}_n^T \otimes I_2 \end{bmatrix} f = \mu mg \begin{bmatrix} -(\chi_4^{ss})^T T\bar{r} \\ \chi_4^{ss} \end{bmatrix}. \tag{4.24}$$

The particular solution in this case is:

$$\begin{aligned} f = f_{tran}^x &= \frac{\mu mg}{n} \left(\begin{bmatrix} 1 & 0 \end{bmatrix} \chi_4^{ss} \right) \left(\mathbb{1}_n \otimes \begin{bmatrix} 1 & 0 \end{bmatrix}^T \right) \\ f = f_{tran}^y &= \frac{\mu mg}{n} \left(\begin{bmatrix} 0 & 1 \end{bmatrix} \chi_4^{ss} \right) \left(\mathbb{1}_n \otimes \begin{bmatrix} 0 & 1 \end{bmatrix}^T \right), \end{aligned} \tag{4.25}$$

where f_{tran}^x and f_{tran}^y correspond to translations in the $\hat{x}$ and $\hat{y}$ direction respectively.

Proposition 4.3.1. *In the general case, where the positioning of the robots is not necessarily symmetric about the center of mass, a pure translation in the $\hat{x}$ or $\hat{y}$ direction can be achieved in steady state, by applying the force described in* (4.25) *through the robots, regardless of the initial conditions.*

Proof. To prove this claim, we need to substitute the proposed forces into the steady state dynamics and show that the resulting vector corresponds to the desired motion, depicted in (4.25).

$$M f_{tran}^x = \begin{bmatrix} r^T (I_n \otimes T) \\ (1_n^T \otimes I_2) \end{bmatrix} \frac{\mu mg}{n} \left(\begin{bmatrix} 1 & 0 \end{bmatrix} \chi_4^{ss} \right) \left(1_n \otimes \begin{bmatrix} 1 & 0 \end{bmatrix}^T \right) \tag{4.26}$$

$$= \mu mg \begin{bmatrix} -(\chi_4^{ss})^T T\bar{r} \\ \chi_4^{ss} \end{bmatrix} \tag{4.27}$$

$$Mf_{tran}^{y} = \begin{bmatrix} r^T (I_n \otimes T) \\ (\mathbb{1}_n^T \otimes I_2) \end{bmatrix} \frac{\mu mg}{n} \left(\begin{bmatrix} 0 & 1 \end{bmatrix} \chi_4^{ss} \right) \left(\mathbb{1}_n \otimes \begin{bmatrix} 0 & 1 \end{bmatrix}^T \right) \tag{4.28}$$

$$= \mu mg \begin{bmatrix} -(\chi_4^{ss})^T T \bar{r} \\ \chi_4^{ss} \end{bmatrix} \tag{4.29}$$

where χ_4^{ss} is the desired steady state linear velocity. The forces f_{tran}^x and f_{tran}^y indeed produce the expected outcome, concluding this proof. □

4.4 Connection to Rigidity Theory

One of the main goals of this work, is to establish a connection between rigidity theory, and the forces needed to move an object utilizing cooperating robots. The idea is to find a simple way to compute the desired forces (instead of going through the dynamics of the system), and in this section we demonstrate that this simpler way can be achieved by utilizing tools from rigidity theory.

After analyzing the motion matrix of the system, we found the forces required in order to create ss translations and rotations, and also the forces for equilibrium. Next we show that the translation and rotation forces are in the null space of the rigidity matrix $R(p)$:

Proposition 4.4.1. *The forces $f_{tran}^x, f_{tran}^y, f_{rot}^z$ are in the null-space of $R(p)$.*

Before we prove this proposition, we need the following useful result relating the rigidity matrix $R(p)$ to $R(r)$.

Proposition 4.4.2. *The rigidity matrix $R(p)$ can be expressed in terms of the relative position vectors r. That is,*

$$R(p) = R(r) \tag{4.30}$$

Proof. The relative position of robot i with respect to robot j is described as:

$$p_{ij} = p_i - p_j, \tag{4.31}$$

in addition, the position of robot i can be expressed in terms of the COM:

$$p_i = p_S + r_i. \tag{4.32}$$

As a result:

$$\begin{aligned} p_{ij} &= p_i - p_j \\ &= p_S + r_i - p_S - r_j \\ &= r_i - r_j \\ &= r_{ij} \end{aligned} \tag{4.33}$$

In short $p_{ij} = r_{ij}$. Note that $R(p)$ is composed of relative positions, that is, p_{ij}. Also the positions of the robots can be described by the position of the center of mass $p_i = p_S + r_i$. Consequently we can also write the rigidity matrix in terms of the position vector r.□

With this result in hand, we can continue to prove Proposition 4.4.1:

Proof. The translation forces can be rearranged as follows:

$$\begin{aligned} f^x_{tran} &= \frac{\mu m g}{n}\left(\begin{bmatrix} 1 & 0 \end{bmatrix}\chi^{ss}_4\right)\left(\mathbb{1}_n \otimes \begin{bmatrix} 1 & 0 \end{bmatrix}^T\right) = \frac{\mu m g}{n}\left(\begin{bmatrix} 1 & 0 \end{bmatrix}\chi^{ss}_4\right) u_1 \\ f^y_{tran} &= \frac{\mu m g}{n}\left(\begin{bmatrix} 0 & 1 \end{bmatrix}\chi^{ss}_4\right)\left(\mathbb{1}_n \otimes \begin{bmatrix} 0 & 1 \end{bmatrix}^T\right) = \frac{\mu m g}{n}\left(\begin{bmatrix} 0 & 1 \end{bmatrix}\chi^{ss}_4\right) u_2. \end{aligned} \tag{4.34}$$

where u_1 and u_2 are the trivial motions described in (2.34). Clearly f^x_{tran} is proportional to u_1 and f^y_{tran} is proportional to u_2, thus f^x_{tran} and f^y_{tran} are in the null space of $R(p)$. Regarding f^z_{rot}, we need some more steps. Recall the intuitive definition of the rigidity matrix and the example of (2.32). Next, let's have a detailed look at the rows of the rigidity matrix. The matrix has one row for each edge in the graph, and a pair of columns for each node. So we can associate the entries of the matrix to a certain edge and node. For instance, a row that corresponds to an edge that connects nodes i and j will look like this:

$$\mathcal{R}_{ij}(p) = \begin{bmatrix} \cdots & p^T_{ij} & \cdots & -p^T_{ij} & \cdots \end{bmatrix}, \tag{4.35}$$

note that p^T_{ij} is in columns $2i-1$ and $2i$, and $-p^T_{ij} = p^T_{ji}$ is in columns $2j-1$ and $2j$, other than these entries the row is filled with zeroes. This property helps us see that $\mathcal{R}_{ij}(p)u_1 = \mathcal{R}_{ij}(p)u_2 \equiv 0$, and if we rewrite the rigidity matrix in terms of the relative positions, then also $\mathcal{R}_{ij}(r)u_3(r) \equiv 0$. □

The meaning of all this, is that we can span both the null space of the rigidity matrix, and the image of the transposed motion matrix with the same 3 vectors. Namely:

Lemma 4.4.3. *Given an infinitesimally rigid framework in* $\mathbb{R}^2$*, and a system with the dynamics described in* (3.3)*, such that the motion matrix of the system is* M *as introduced in* (4.7)*; the null space of the rigidity matrix is the image of the transposed motion matrix of the system:*

$$Null\{R(r)\} = Im\left\{M^T\right\}, \tag{4.36}$$

furthermore:

$$Null\{M\} = Im\left\{R^T(r)\right\}. \tag{4.37}$$

Proof. Within this context, the null space of the rigidity matrix can be spanned by the following vectors:

$$Null\{R(r)\} = span\left\{u_1, u_2, u_3(r)\right\}, \tag{4.38}$$

where

$$u_1 = \mathbb{1}_n \otimes \begin{bmatrix} 1 & 0 \end{bmatrix}^T \tag{4.39}$$

$$u_2 = \mathbb{1}_n \otimes \begin{bmatrix} 0 & 1 \end{bmatrix}^T \tag{4.40}$$

$$u_3(r) = (I_n \otimes T)\, r. \tag{4.41}$$

Note that these 3 vectors produce all possible trivial motions when applied as forces to the system. That is, *any* motion can be produced by applying a linear combination of u_1, u_2 and $u_3(r)$. Mathematically speaking, for some constants $c_1, c_2, c_3 \in \mathbb{R}$: $f = c_1u_1 + c_2u_2 + c_3u_3(p)$, the product Mf can be any $B \in \mathbb{R}^3$. In other words:

$$Im\{M^T\} = span\,\{u_1, u_2, u_3(r)\} = Null\{R(r)\}, \tag{4.42}$$

and by the fundamental theorem of linear algebra:

$$\mathrm{Null}\,\{M\} = \mathrm{Im}\,\left\{R^T(r)\right\}. \tag{4.43}$$

This conclude our proof. □

This result is very interesting because it shows a connection between rigidity theory and the dynamics of an object to be moved by a group of cooperating robots. It shows us, in particular, that when looking for the forces required to create basic motions (i.e. translations and rotations), there is no need to compute the dynamics of the system. Instead, we may compute the rigidity matrix of the corresponding underlying graph, and find the needed forces from its null space.

In detail, for a group of robots that are tasked with moving an object in a cooperative manner, the forces required to create the basic motions are the null space vectors of the corresponding rigidity matrix. That is, the translation force is $u_1 = \mathbb{1} \otimes \begin{bmatrix} 1 & 0 \end{bmatrix}^T$ (in other words each robots applies the force $f_i = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$) if movement in the $\hat{x}$ direction is required, $u_2 = \mathbb{1} \otimes \begin{bmatrix} 0 & 1 \end{bmatrix}^T$ (in other words each robots applies the force $f_i = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$) if movement in the $\hat{y}$ direction is required, and the rotation force is $u_3(r) = (I_n \otimes T)\, r$ (in other words each robot applies the force $f_i = Tr_i$).

Chapter 5

Simulation Results

In this chapter we analyze the results of a *Matlab* simulation, where the system presented in (3.3) is subject to the various forces of interest. The parameters of the simulation are the following: The mass of the system is $m = 32.7\,[kg]$. The $\hat{z}$ moment of inertia of the system about the center of mass is $\mathrm{I} = 52.0447\,\left[kg \cdot m^2\right]$. The friction coefficient between the robots and the floor is $\mu = 0.25$. The gravitational constant is $g = 9.81\,\left[\frac{m}{s^2}\right]$. The number of robots is $n = 4$, and the initial (asymmetric) position of the robots is

$$p\,(0) = \left[\begin{array}{cccccccc} -0.0303 & 0.3459 & 0.1697 & 0.3459 & -0.2303 & 0.1459 & -0.2303 & -0.6541 \end{array}\right]^T.$$

In this simulation the required forces that the robots have to apply are computed via the null space of the rigidity matrix of the framework, and these forces are then substituted as an input in the dynamics of the system (3.3).

5.1 Translation Forces

Here, the forces applied to the system correspond to the vectors u_1 and u_2, from two initial conditions: rest and motion, in the general case where the positioning of the robots is not symmetric. The variables of the simulation are as follows: The applied force is $f = \mathbb{1}_n \otimes \left[\begin{array}{cc} 1 & 0 \end{array}\right]^T$, and the initial conditions are

$$\begin{aligned} \chi_1\,(0) &= 0 \\ \chi_2\,(0) &= 0 \\ \chi_3\,(0) &= \left[\begin{array}{cc} 0 & 0 \end{array}\right]^T \\ \chi_4\,(0) &= \left[\begin{array}{cc} 0 & 0 \end{array}\right]^T. \end{aligned} \tag{5.1}$$

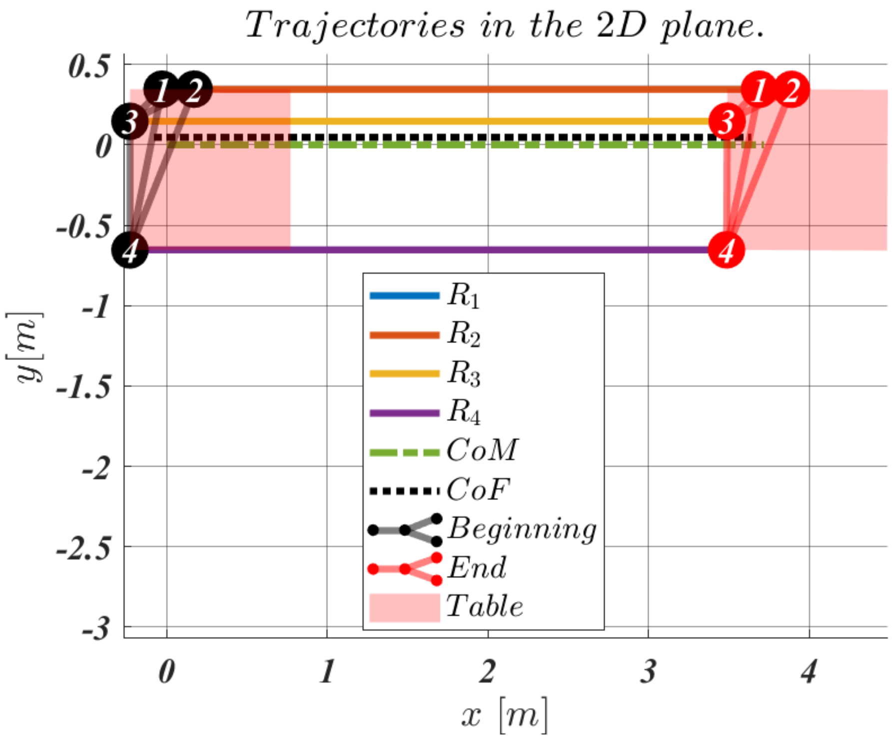
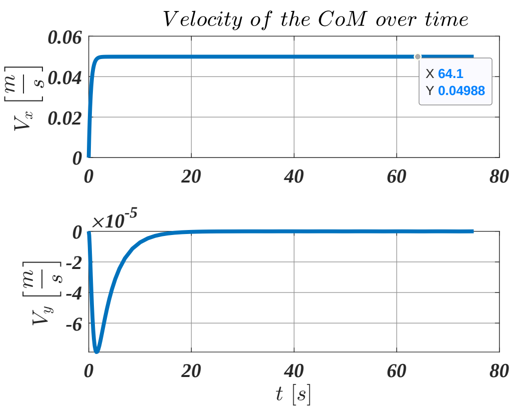
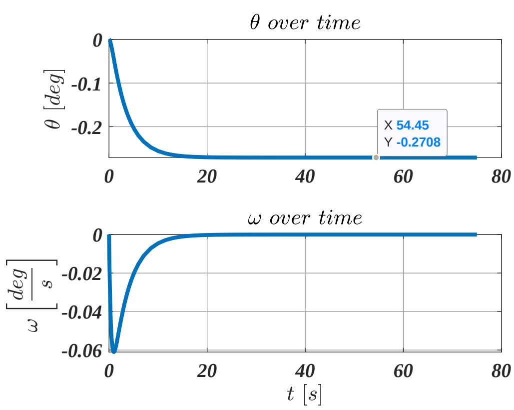


Figure 5.1: The planar motion caused by applying the force u_1 on the table. *CoM* marks the center of mass of the system, which in this asymmetric case does not coincide with the centroid of the formation *CoF*.



(a) The linear velocity of the CoM reaches a steady state in the $\hat{x}$ direction.



(b) The angular velocity and orientation of the table.

Figure 5.2: Trajectories created by the force u_1 on the table, the initial conditions are zero.

Figures 5.1, 5.2a and 5.2b show that after the transient dies out, the steady state velocity is in the $\hat{x}$ direction.

And when the force is $f = \mathbb{1}_n \otimes \begin{bmatrix} 0 & 1 \end{bmatrix}^T$ as expected the results are not that different:

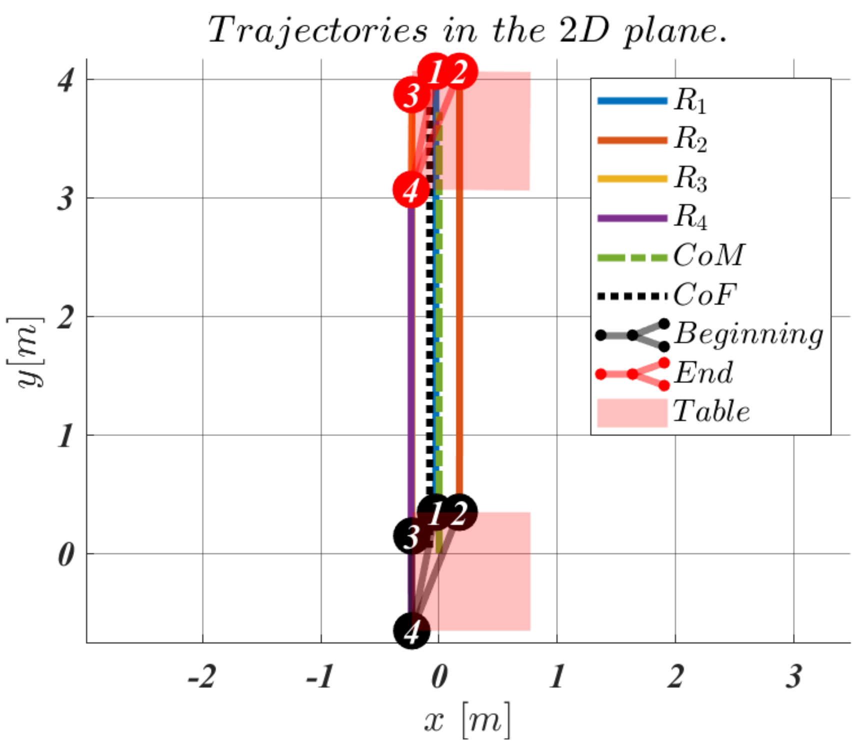
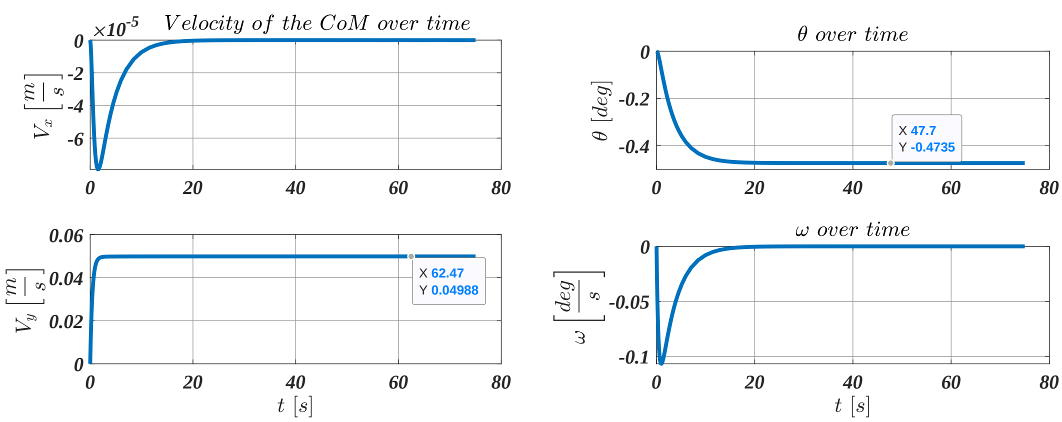


Figure 5.3: The planar motion caused by applying the force u_2 on the table. *CoM* marks the center of mass of the system, which in this asymmetric case does not coincide with the centroid of the formation *CoF*.



(a) The linear velocity of the CoM reaches a steady state in the $\hat{y}$ direction.

(b) The angular velocity and orientation of the table.

Figure 5.4: Trajectories created by the force u_2 on the table, the initial conditions are zero.

These results are similar to the ones just presented, which makes sense, since the only difference was the direction of the desired translation.
Now we change the initial conditions, and repeat the process. That is, apply the force

u_1 and then u_2, with the following initial conditions:

$$\chi(0) = \begin{bmatrix} 0 & 30 & 00 & 00 \end{bmatrix}^T \tag{5.2}$$

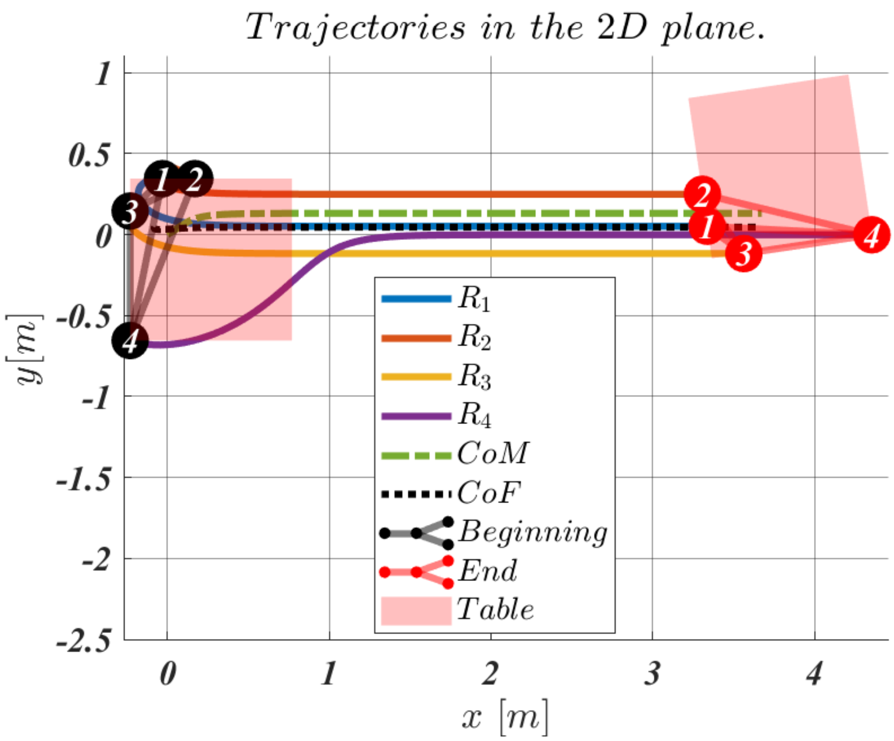
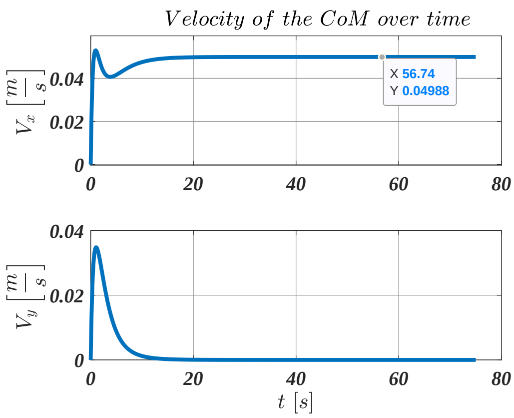
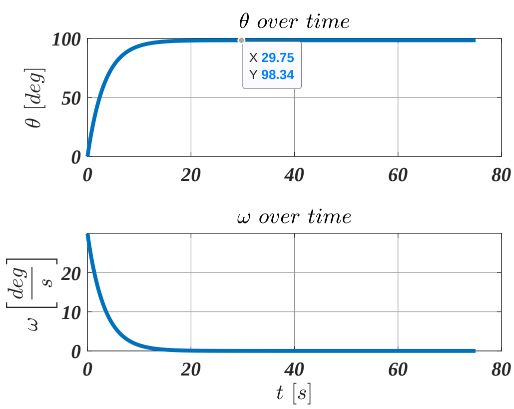


Figure 5.5: The planar motion caused by applying the force u_1 on the table. This simulation began with the table spinning.



(a) The linear velocity of the center of mass reaches a steady state in the $\hat{x}$ direction, despite the initial spin.



(b) The angular velocity is stabilized in spite of the initial spin.

Figure 5.6: Trajectories created by the force u_1 on the table, with an initial spin of $30\,[deg/s]$ as initial conditions.

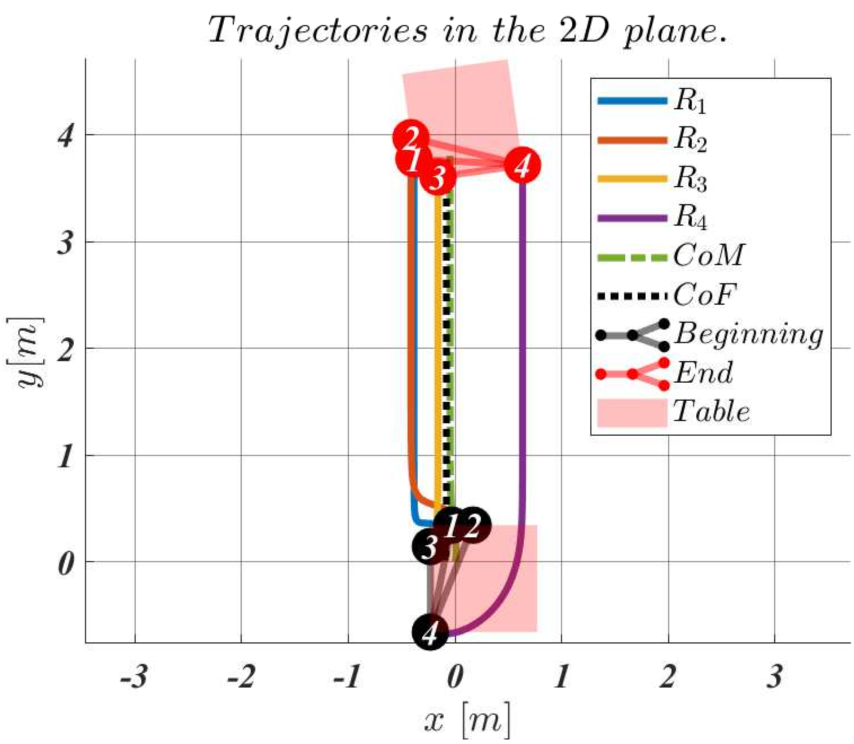
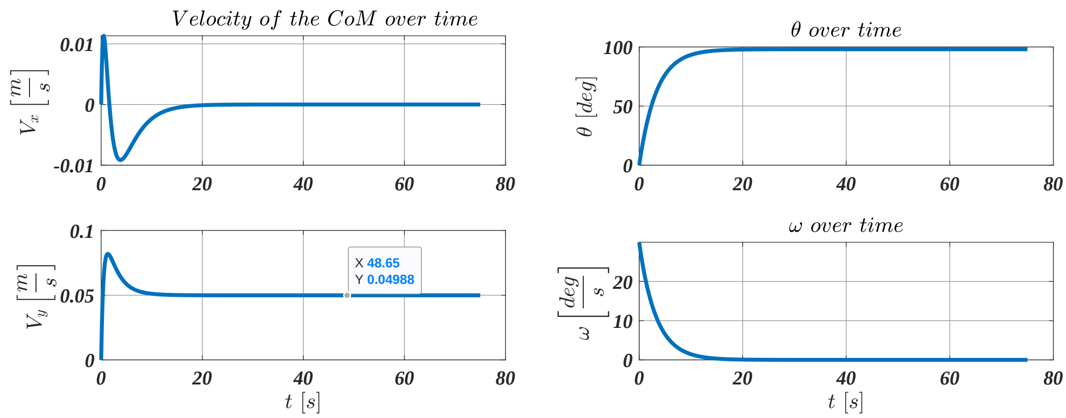


Figure 5.7: The planar motion caused by applying the force u_2 on the table. *CoM* marks the center of mass of the system, which in this asymmetric case does not coincide with the centroid of the formation *CoF*.



(a) The linear velocity of the CoM reaches a steady state in the $\hat{y}$ direction.

(b) The angular velocity and orientation of the table.

Figure 5.8: The linear velocity of the center of mass reaches a steady state in the $\hat{y}$ direction.

Figures 5.5, 5.6a and 5.6b show that an initial rotation does not stop the system from reaching a steady state translation. This can also be seen in the case of Figures 5.7, 5.8a and 5.8b, which present motion in the $\hat{y}$ direction.

Now, if instead of an initial rotation we apply an initial linear velocity:

$$\chi(0) = \begin{bmatrix} 0 & 0 & 00 & \sqrt{\pi}\sqrt{e} \end{bmatrix}^T \tag{5.3}$$

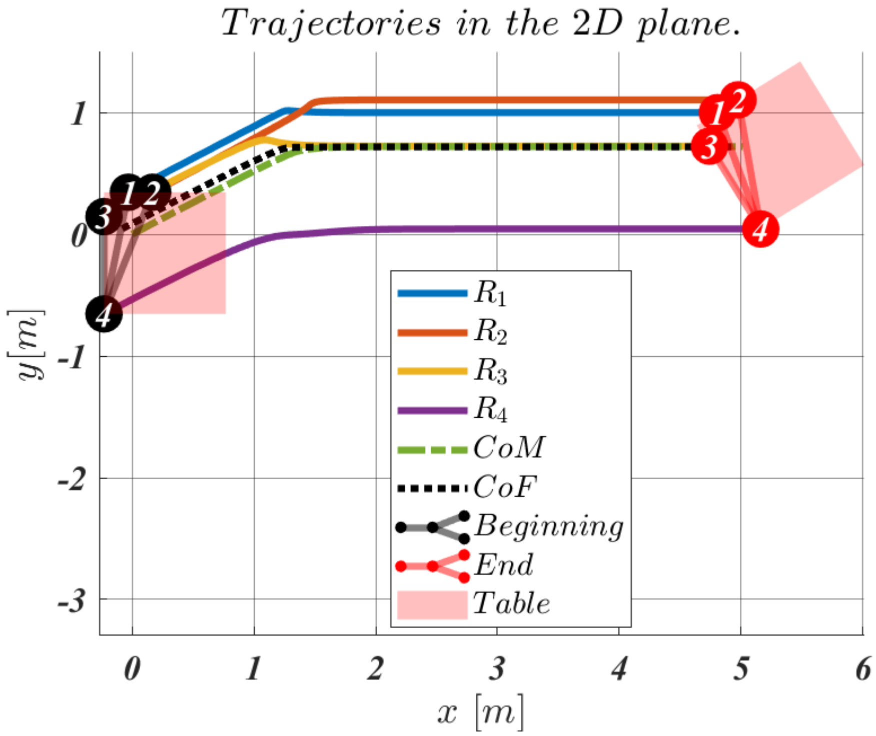
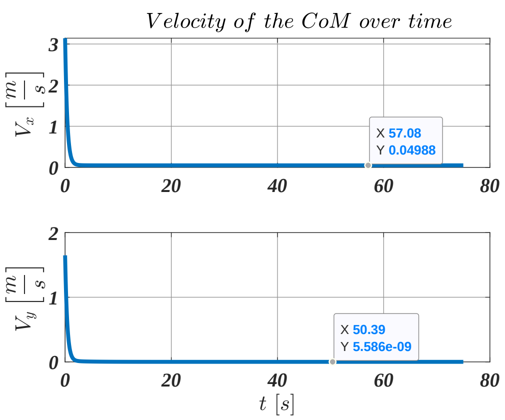
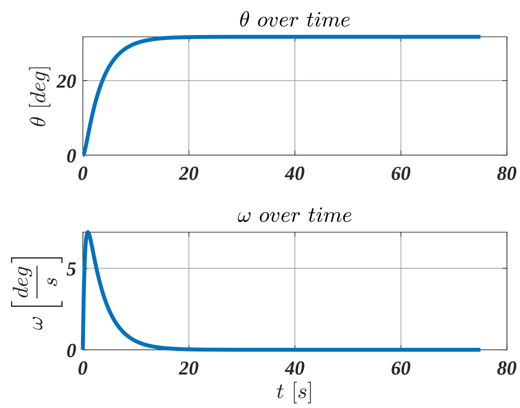


Figure 5.9: The planar motion caused by applying the force u_1 on the table. This simulation began with the table shifting.



(a) The linear velocity of the CoM reaches a steady state in the $\hat{x}$ direction.



(b) The angular velocity is stabilized in spite of the initial shift.

Figure 5.10: An initial shift does not prevent the angular velocity from decaying, allowing the system to reach a steady state with a linear velocity.

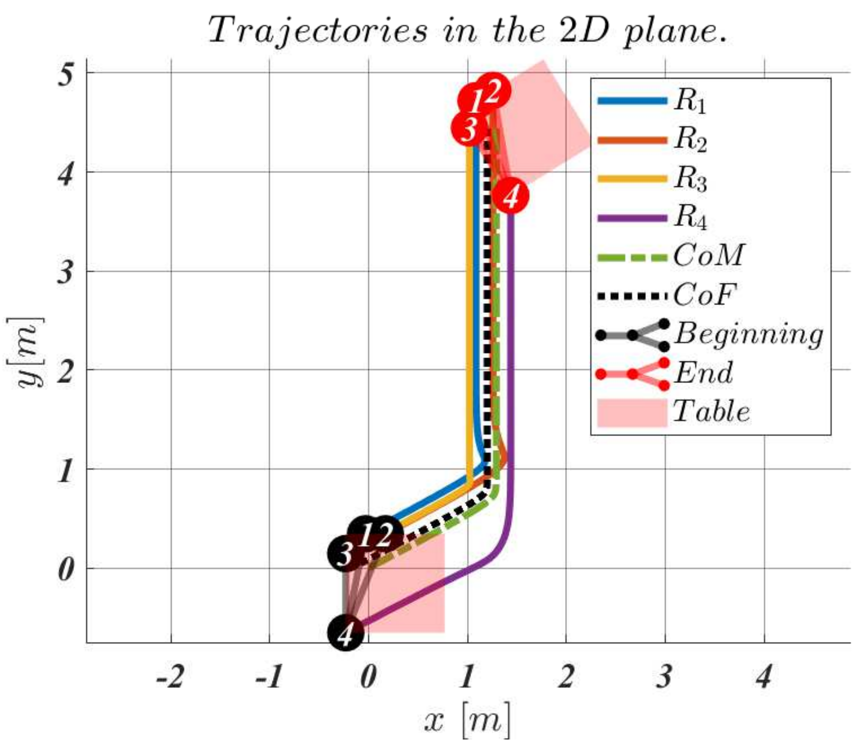
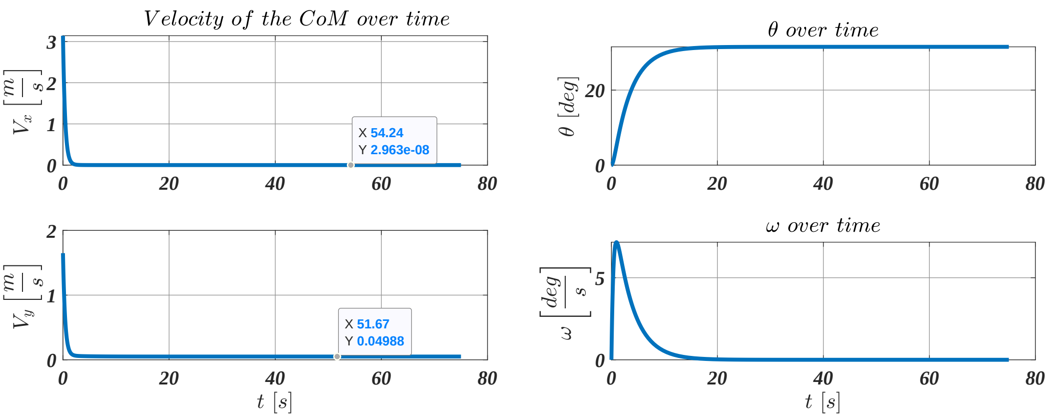


Figure 5.11: The planar motion caused by applying the force u_2 on the table. *CoM* marks the center of mass of the system, which in this asymmetric case does not coincide with the centroid of the formation *CoF*.



(a) The linear velocity of the CoM reaches a steady state in the $\hat{y}$ direction.

(b) The angular velocity and orientation of the table.

Figure 5.12: Here the initial conditions are initial velocity, and the outcome is a steady state translation with no rotation.

Figures 5.9, 5.10a and 5.10b show that an initial shift does not stop the system from reaching a steady state translation, as can also be seen in Figures 5.11, 5.12a and 5.12b.

5.2 Rotation Forces

In this section we repeat the sequence of simulations, with the only difference being the applied force. The initial conditions are the same as in (5.1), and the applied force is $f = u_3(p) = (I_n \otimes T)\, p$.

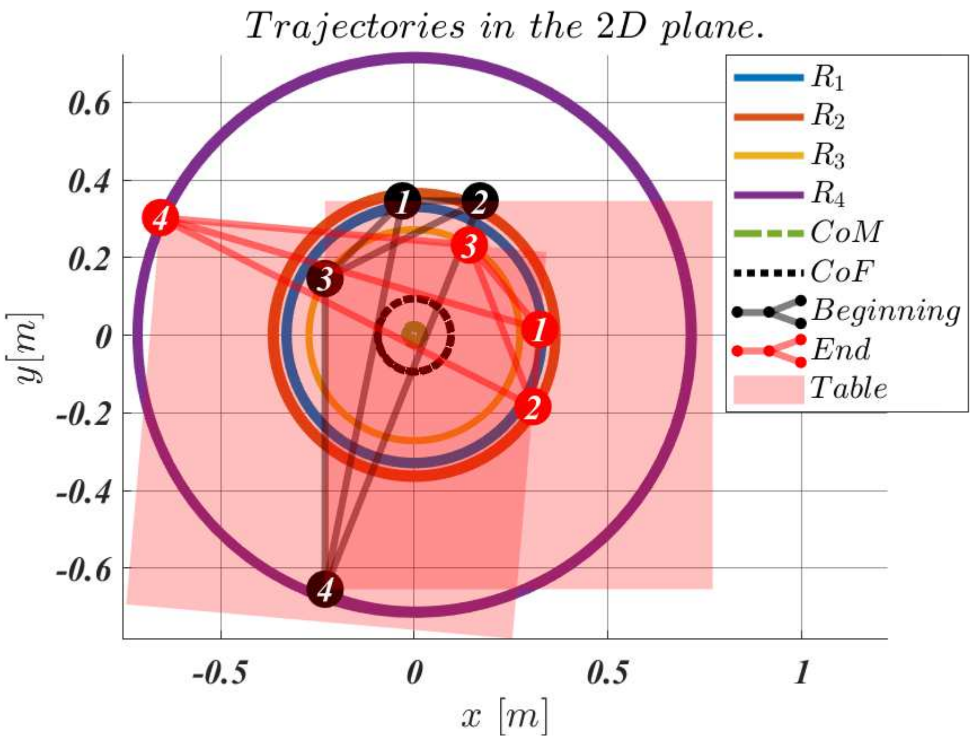
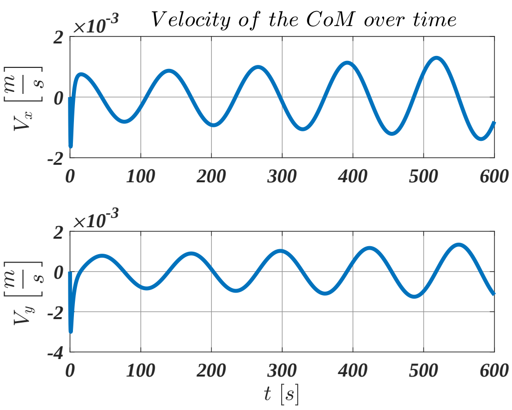
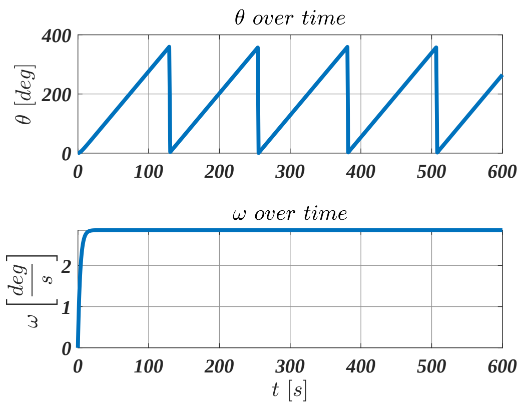


Figure 5.13: The planar motion caused by applying the force $u_3(p)$ on the table. CoM marks the center of mass of the system, which in this asymmetric case does not coincide with the centroid of the formation CoF.



(a) The linear velocity of the CoM does not converge.



(b) The angular velocity and orientation of the table.

Figure 5.14: The outcome of applying the force $u_3(p)$ is a linear combination of rotation and translations.

Clearly this force does not create a pure rotation. This can be seen in Figure 5.14a, where the linear velocity is shown to be fluctuating. We now change the initial conditions to the ones depicted in (5.2).

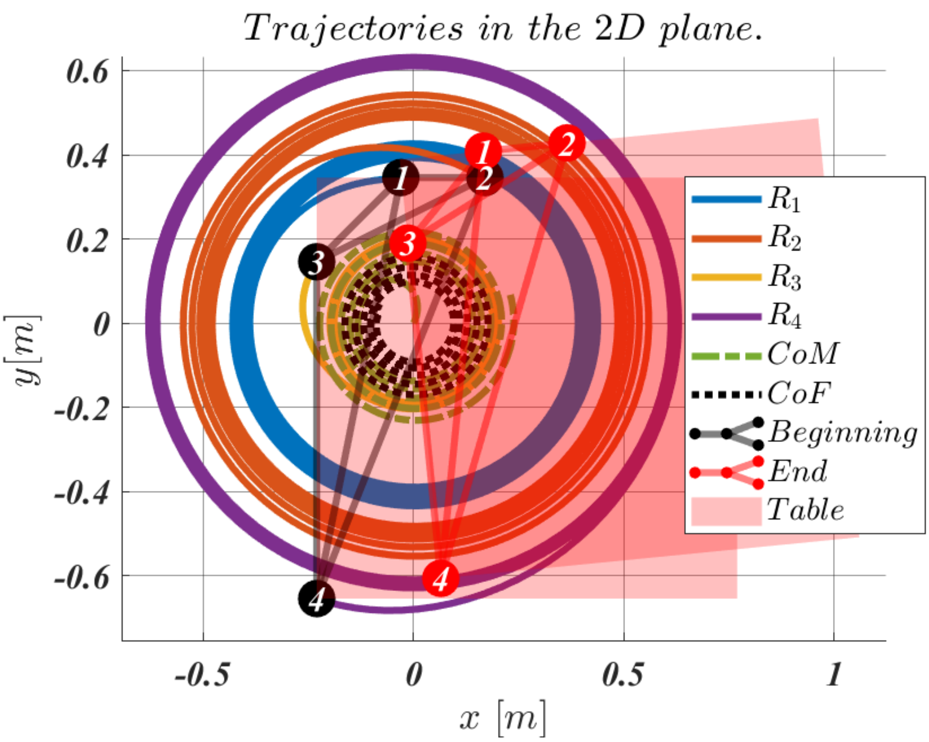
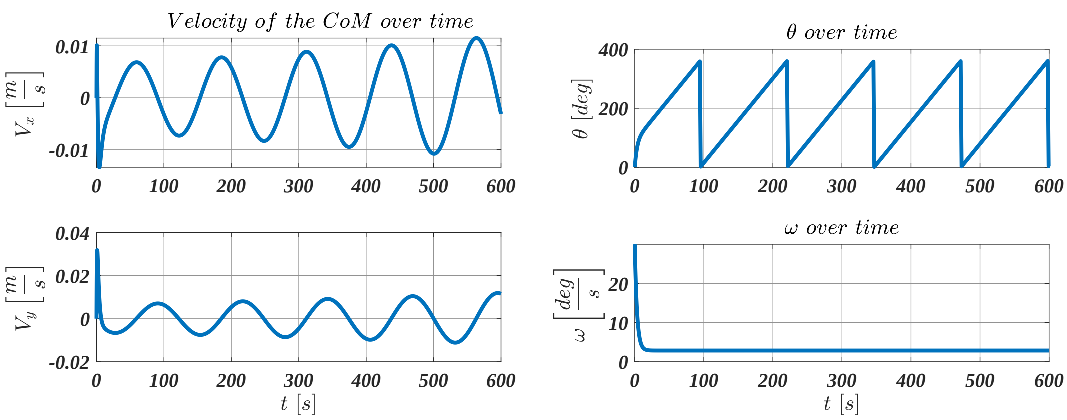


Figure 5.15: The planar motion caused by applying the force $u_3(p)$ on the table. CoM marks the center of mass of the system, which in this asymmetric case does not coincide with the centroid of the formation CoF.



(a) The linear velocity of the CoM does not converge.

(b) The angular velocity and orientation of the table.

Figure 5.16: An initial spin does not seem to affect the uncontrollable fluctuations in the linear velocity.

And in the case of (5.3) (linear velocity as initial conditions):

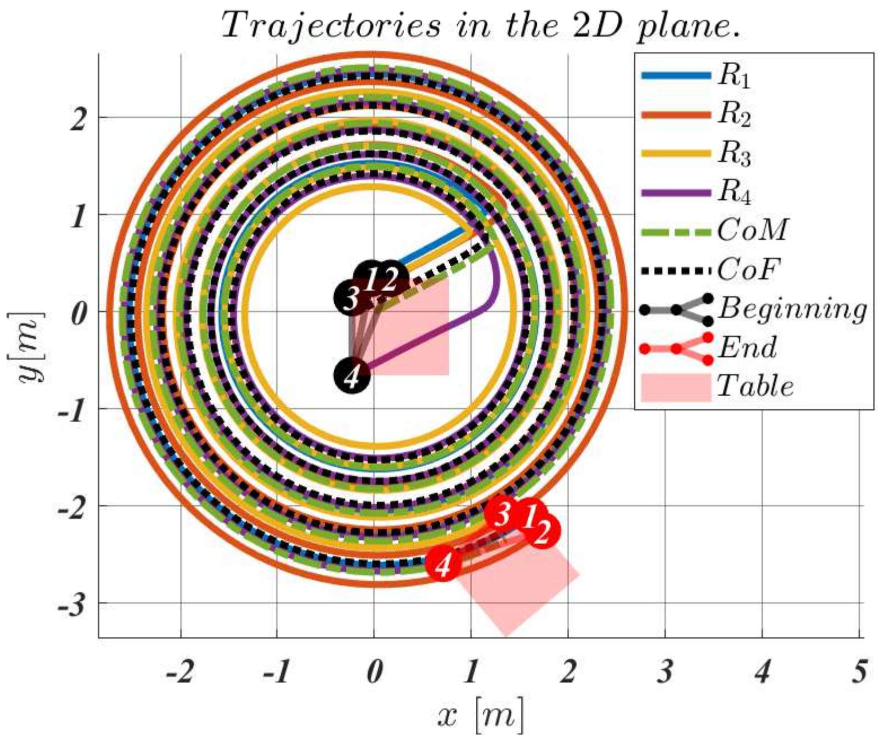
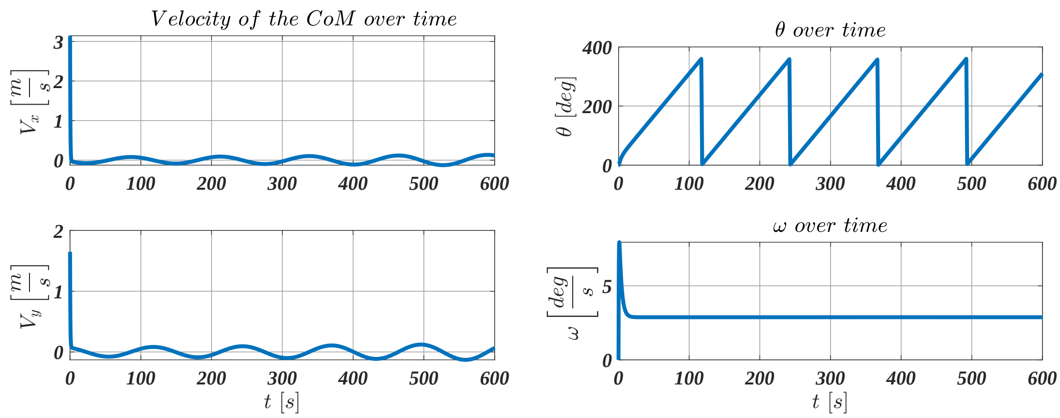


Figure 5.17: The planar motion caused by applying the force $u_3(p)$ on the table. CoM marks the center of mass of the system, which in this asymmetric case does not coincide with the centroid of the formation CoF.



(a) The linear velocity of the CoM does not converge.

(b) The angular velocity and orientation of the table.

Figure 5.18: When the system starts with a shift, the linear velocity does not converge.

The result that can be seen from these figures is that the initial conditions do not help in the general case with an asymmetric positioning.
Now we're in a position to compare the results to the ones obtained by the force $f = u_3(r) = (I_n \otimes T)\, r$. Initial conditions - (5.1)

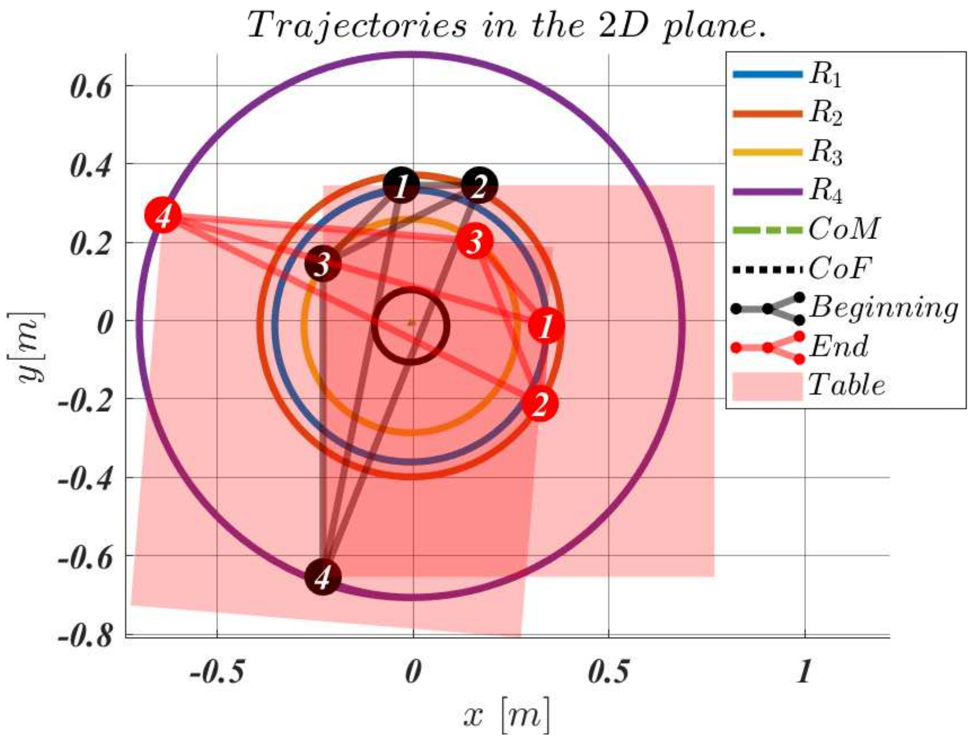
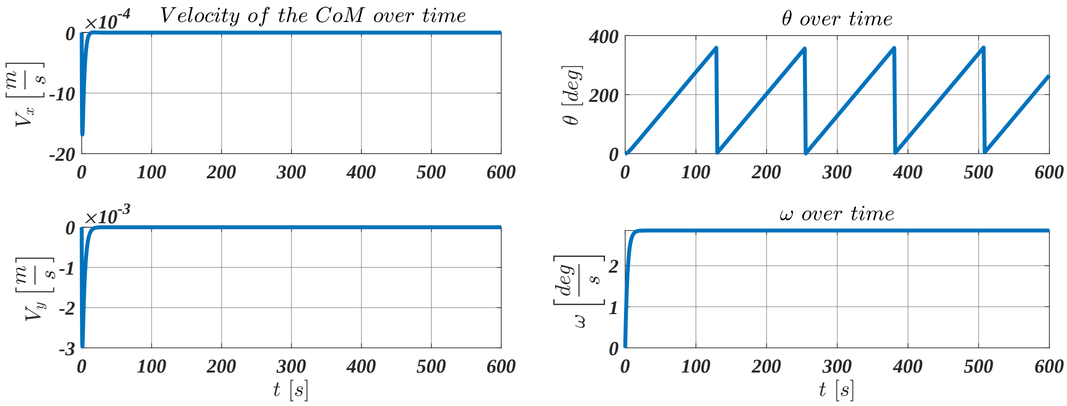


Figure 5.19: The planar motion caused by applying the force $u_3(r)$ on the table. CoM marks the center of mass of the system, which in this asymmetric case does not coincide with the centroid of the formation CoF.



(a) The linear velocity of the CoM converges to zero.

(b) The angular velocity and orientation of the table.

Figure 5.20: As expected, from zero initial conditions the forces $u_3(p)$ and $u_3(r)$ produce the same output.

These figures show that from zero initial conditions, the force $u_3(r)$ causes a steady state rotation about the formation's centroid. Next we'll see the effect of other initial conditions on the outcome of this force.

When starting with an initial rotation (5.2):

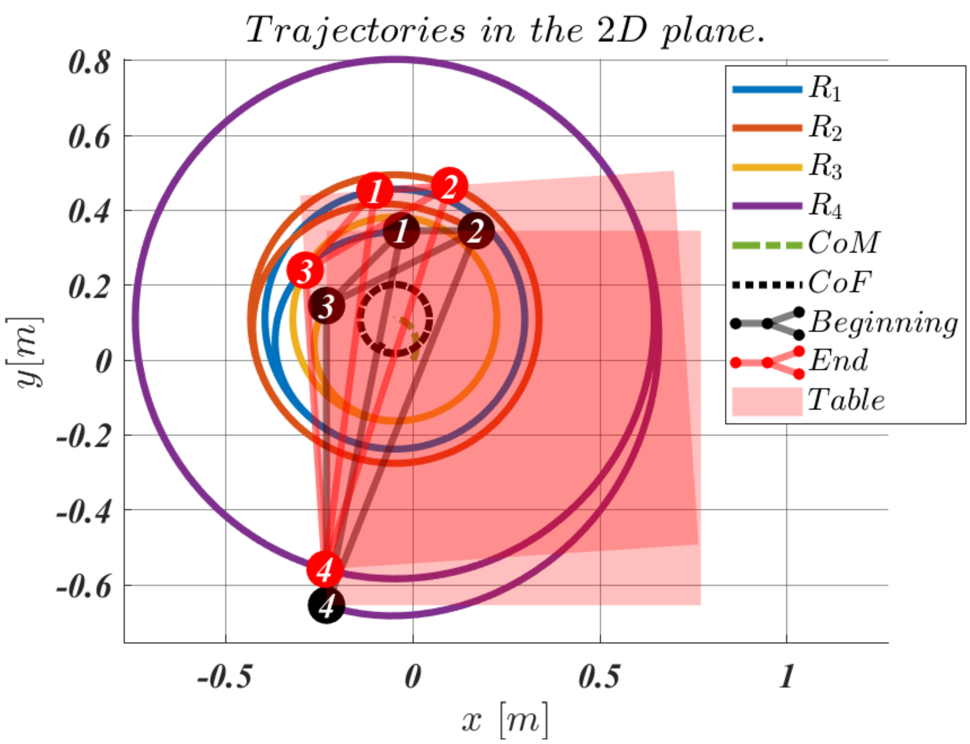
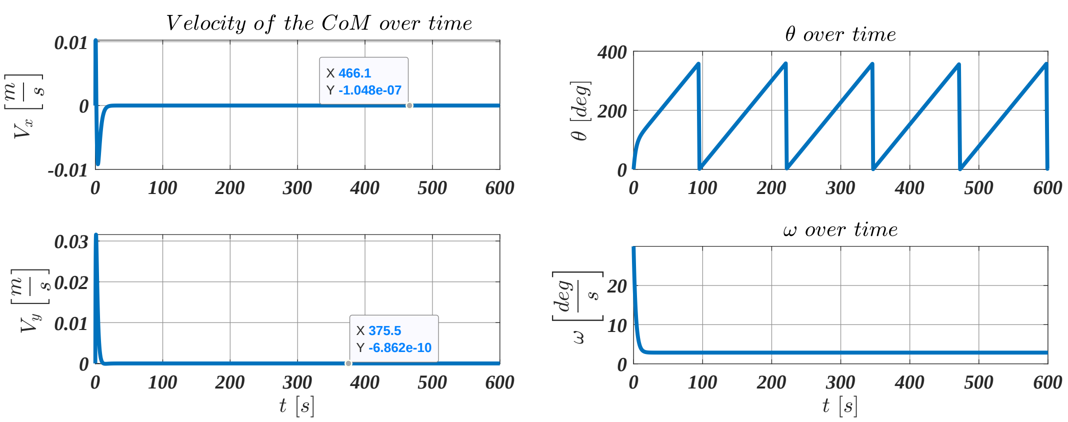


Figure 5.21: The planar motion caused by applying the force $u_3(r)$ on the table. CoM marks the center of mass of the system, which in this asymmetric case does not coincide with the centroid of the formation CoF.



(a) The linear velocity of the CoM converges to zero.

(b) The angular velocity and orientation of the table.

Figure 5.22: The initial spin does not prevent the table from reaching a steady state rotation with no translations.

We can learn form these figures that when the system starts with an angular velocity of 30 $[deg/s]$ as initial conditions, a steady state rotation is achieved. When the initial conditions are a linear velocity (5.3):

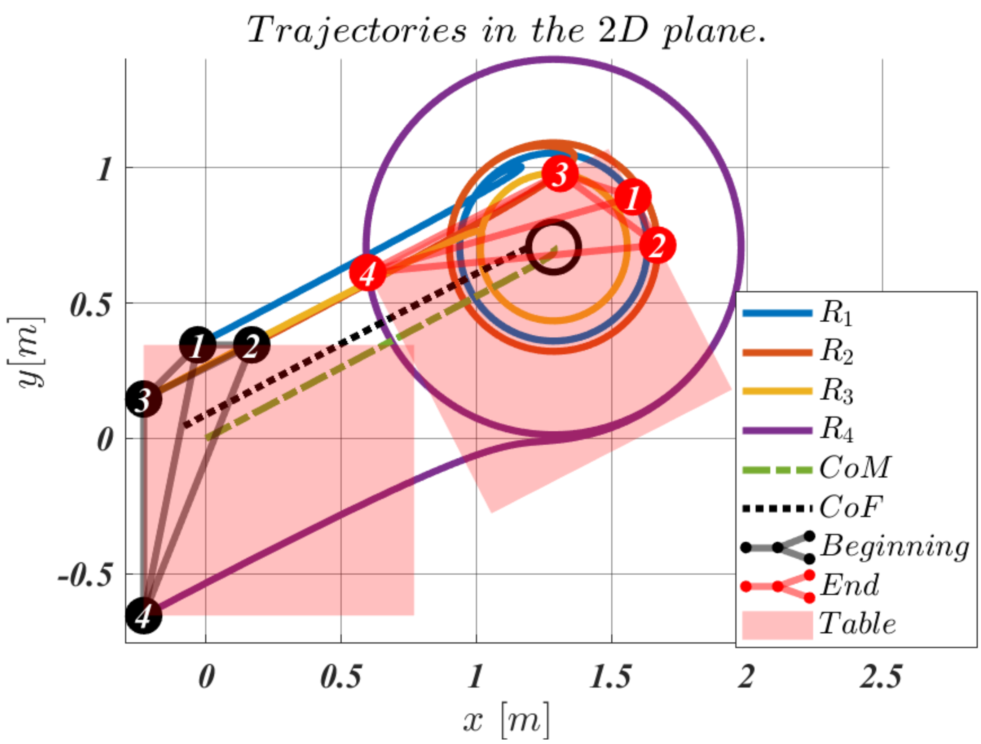
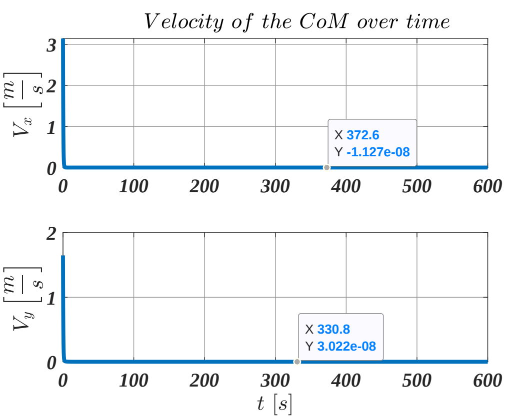
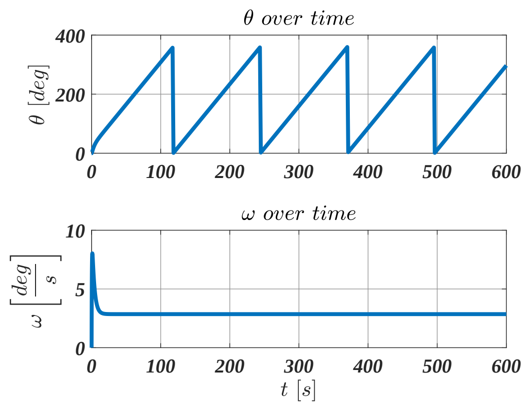


Figure 5.23: The planar motion caused by applying the force $u_3(r)$ on the table. *CoM* marks the center of mass of the system, which in this asymmetric case does not coincide with the centroid of the formation *CoF*.



(a) The linear velocity of the CoM converges to zero.



(b) The angular velocity and orientation of the table.

Figure 5.24: Even with an initial shift as initial conditions, the table reaches a steady state with a constant angular velocity, and no translations.

Figures 5.24a, 5.24b and 5.23 point out that even with an initial linear velocity, the force $u_3(r)$ causes a steady state rotation.

Concluding these results, we've seen that both in the symmetric and asymmetric cases, the forces u_1 and u_2 produce steady state translations, regardless of the initial conditions. The force $u_3(p)$ causes a steady state rotation in the symmetric case, but only from zero initial conditions (this was explained analytically in (3.17), and the simulation results shown in Figure 3.6). In the asymmetric case the force $u_3(p)$ causes a linear combination of translations and rotations, from all kinds of initial conditions (rest, initial rotation or linear velocity). On the other hand, the force $u_3(r)$ causes a steady state rotation regardless of the initial conditions, or robots positioning - corroborating the analytical results presented in Chapter 4.

Chapter 6

Summary and Future Research

6.1 Research Summary

This research started by analyzing the dynamics of an object moved in the plane by multiple manipulators. Since a group of agents apply forces at the same time, graph theory was utilized to describe the interaction between those agents. Rigidity theory was connected to the forces needed to move the object through the rigidity matrix of the corresponding framework, and it was shown that the null space of the rigidity matrix is the range of the transposed motion matrix of the system. In other words to find the required forces to move the object in the very basic motions (steady state translations and rotations), we need only to compute the null space vectors of the rigidity matrix, and not necessarily go through the dynamics of the system. Finally these forces were tested in simulation and the results were corroborated.

6.2 Future Research

Extend to $\mathbb{R}^3$: Even though a lot of applications require in-plane motion, it would be beneficial to develop the connection between the rigidity and motion matrices in the $3D$ case. In such a scenario the friction should be modeled differently, the object would tilt in flight, and these are just some of the changes that should be addressed in terms of the dynamics. In addition, for the rigidity matrix to be of full column rank $(3n-6)$ the formation should not be co-planar, and this poses yet another requirement to be met when planning the positioning of the robots around the cargo.

Moving cargo : In the existing analysis the cargo to be transported is modeled as a non-moving object. In reality perhaps animals or liquids in tanks are moved, and could induce disturbances that should be modeled as well. Of course this opens the door to developing a control strategy to move payloads from a to b, which is yet another action item to be dealt with in future research.

Bibliography

[1] S. Akella and M. T. Mason. Posing polygonal objects in the plane by pushing. *The International Journal of Robotics Research*, 17(1):70–88, 1998.

[2] A. Alfakih. On the dual rigidity matrix. *Linear Algebra and its Applications*, 428(4):962 – 972, 2008.

[3] Brown university, division of engineering. Dynamics and vibrations. `http://www.brown.edu/Departments/Engineering/Courses/En4/notes_old/RigidKinematics/rigkin.htm`. [Online; accessed 15-July-2018].

[4] R. Connelly. Generic global rigidity. *Discrete & Computational Geometry*, 33(4):549–563, Apr 2005.

[5] A. Franchi, A. Petitti, and A. Rizzo. Distributed estimation of state and parameters in multi-agent cooperative manipulation. *IEEE Trans. on Control of Network Systems*, 2018.

[6] C. Godsil and G. Royle. *Algebraic Graph Theory*. Springer, 2001.

[7] K. M. Lynch. The mechanics of fine manipulation by pushing. In *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pages 2269–2276 vol.3, May 1992.

[8] M. Mesbahi and M. Egerstedt. *Graph Theoretic Methods in Multiagent Networks*. Princeton Series in Applied Mathematics. Princeton University Press, 2010.

[9] G. A. S. Pereira, V. Kumar, and M. F. M. Campos. Closed loop motion planning of cooperating mobile robots using graph connectivity. *Robotics and Autonomous Systems*, 56(4):373 – 384, 2008.

[10] B. Roth. Rigid and flexible frameworks. *The American Mathematical Monthly*, 88(1):6–21, 1981.

[11] T. Sakuyama, J. David Figueroa Heredia, T. Ogata, T. Hara, and J. Ota. Object transportation by two mobile robots with hand carts. 2014:1–15, 10 2014.

[12] M. Stilman and J. Kuffner. Navigation among movable obstacles: real-time reasoning in complex environments. In *4th IEEE/RAS International Conference on Humanoid Robots, 2004.*, volume 1, pages 322–341 Vol. 1, Nov 2004.

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

[14] T.-S. Tay. Rigidity of multi-graphs. i. linking rigid bodies in n-space. *Journal of Combinatorial Theory, Series B*, 36(1):95 – 112, 1984.

[15] M. Tognon, C. Gabellieri, L. Pallottino, and A. Franchi. Aerial co-manipulation with cables: The role of internal force for equilibria, stability, and passivity. *IEEE Robotics and Automation Letters, Special Issue on Aerial Manipulation*, 3:2577–2583, 02/2018 2018. Also selected for presentation at the 2018 IEEE Int. Conf. on Robotics and Automation, Brisbane , Australia.

[16] M. H. Trinh, M. C. Park, Z. Sun, B. D. O. Anderson, V. H. Pham, and H. S. Ahn. Further analysis on graph rigidity. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 922–927, Dec 2016.

[17] Z. Wang and M. Schwager. Force-amplifying n-robot transport system (force-ants) for cooperative planar manipulation without communication. *The International Journal of Robotics Research*, 35(13):1564–1586, 2016.

[18] Wikipedia contributors. Kronecker product — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Kronecker_product&oldid=848980795`, 2018. [Online; accessed 15-July-2018].

תקציר

בני אדם מזיזים חפצים כבר משחר הזמן. בין אם מדובר בציידים ומלקטים המביאים את הפרס היומי בחזרה לכפר, או עובדי מנהרות המביאים פחם למרכז הסחר הקרוב, המטלה נותרה זהה. היום הצרכים שלנו השתנו ואנו כבר לא צריכים לצוד את האוכל שלנו, אנחנו פשוט הולכים למכולת, אבל עדיין צריך לדחוף את העגלה. אנחנו יכולים לקנות את המוצרים בצורה מקוונת, אבל עדיין נצטרך ללכת לדואר לאסוף אותם. מתחילת הדרך אנו מחפשים פתרונות לבעיות ההובלה שלנו: לדאוג שחיות יחרשו את השדה במקומנו (רעיון אשר אחר כך התחלף בטרקטור), או קרונות של רכבת על מנת להעביר פחם, או רחפנים אשר יעבירו את הקניות המקוונות שלנו מסניף הדואר אל דלת ביתנו.

הסוגיה העיקרית איתה צריך להתמודד כשמנסים להוביל מטענים, היא שכדי להוביל חפץ גדול (או כבד), אנו צריכים חפץ גדול *אחר*. זו אולי לא בעיה באתר בנייה, היכן שמשתמשים בעגורן כדי להזיז משקלים אדירים, אבל זו בהחלט בעיה אם אנחנו רוצים להוביל משהו כבד אל ביתו של לקוח, או אם צריך להזיז את החפץ הכבד בתוך מבנים.

הפתרון שאנו מציעים לא חדש בהיבט הרובוטי, והוא מבוסס על התנהגות בני אדם. כשאנו רוצים להזיז רהיט כבד בבית (נניח מיטה), אנחנו לא משתמשים במלגזה, אלא פשוט מבקשים עזרה מאדם נוסף. מבלי לשים לב אולי, מה שאנו מבקשים מן האדם הנוסף, הוא לחלוק את העומס בצורה מבוזרת. זהו בדיוק העיקרון אשר הוביל אותנו לתיזה של עבודה זו – *להשתמש ברובוטים מרובים על מנת להוביל חפצים, במקום להשתמש רק באחד.*

הדרך הזו להובלת חפצים בעלת יתרונות רבים על השיטות הקונבנציונליות. בתור התחלה, ניתן להשתמש באותה טכניקה כדי להזיז חפצים בכל הצורות והגדלים, בזמן שמלגזות למשל, מקוטלגות לפי העומס המרבי שלהן. זה אומר שמחסן שעד היום היה צריך להחזיק מלגזות שונות (ויקרות) באתר יוכל יום אחד להחליפן בהרבה רובוטים קטנים (וזולים) אשר עובדים לבד על מנת להזיז עומסים קטנים ומשתפים פעולה בעת הצורך להזיז משהו גדול יותר. יתרון נוסף הוא האמינות של הגישה. ניתן בקלות להחליף רובוט תקול ברובוט אחר בקבוצה, לעומת זאת עגורן תקול מביא בנייה לעצירה מוחלטת. בעזרת אותה דוגמא אפשר לראות שהפתרון החדש יותר טוב מבחינה כלכלית, אך זה לא הכל. היתרון העיקרי הוא המגוון הרחב של היישומים לזה. בעתיד נוכל להחליף עגורנים, מלגזות, רכבות, משאיות ואפילו מטוסים עם נושאים מבוזרים, כאשר ההבדל היחיד בין היישומים יהיה מספר הנושאים.

נכון להיום, להזיז חפצים בעזרת רובוטים זה כבר לא משימה חדשה. יחד עם זאת, המשקל והממדים של המטען לעיתים מגבילים את המטלה. על מנת לפתור מגבלה זו, אנו מציעים להשתמש ברובוטים המשתפים פעולה במטרה להזיז חפצים גדולים. האתגר העיקרי בגישה זו הוא שהרובוטים לא יודעים דבר על הגיאומטריה של החפץ אותו הם מזיזים, אבל כן זמינה להם האינפורמציה היחסית של עצמם (כלומר, המיקום, המהירות וההכוון שלהם). עובדה זו מעודדת אותנו לפתור את הבעיה בעזרת תורת הקשיחות, כלי מתמטי אשר השימוש בו עלה לאחרונה בתחום בקרת מבנים. בהקשר הזה, הצימוד של התנועה המישורית של האובייקט ניתן לפירוק להעתקות וסיבוב טהורים, על ידי שימוש בווקטורים העצמיים של מטריצת הקשיחות של אותה הבעיה. הצגת הבעיה בצורה הזו שימושית מכיוון שהיא נשענת על מידע מקומי בלבד בכדי ליצור כל מסלול רצוי. במילים אחרות, על ידי שימוש במיקום ומהירות של הרובוטים ביחס למרכז המסה של המערכת, אנו מסוגלים לחשב את הכוחות הנדרשים על מנת ליצור העתקות וסיבוב של החפץ הנתון להזזה.

עבודה זו חוקרת איך ניתן להשתמש במטריצת הקשיחות של המערכת על מנת למצוא את הכוחות הנדרשים כדי להעתיק ולסובב חפץ, ובכך לאפשר לרובוטים להזיז את החפץ לכל מקום במישור. לבסוף, אני ממחישים את התוצאות האנליטיות בעזרת סימולציות נומריות.

המחקר בוצע בהנחייתו של פרופ׳ דניאל זלזו, בפקולטה להנדסת אווירונאוטיקה וחלל.

עבודה זו נתמכה ע״י הקרן הישראלית למדע תחת מענק 1490/13.

אני מודה לטכניון על התמיכה הכלכלית הנדיבה בהשתלמותי.

מניפולציה שיתופית של אובייקטים
גישת תורת הקשיחות

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר
מגיסטר למדעים בהנדסת אווירונאוטיקה וחלל

דאגלס גולדנברג

הוגש לסנט הטכניון – מכון טכנולוגי לישראל
תשרי התשע״ט חיפה אוקטובר 2018

מניפולציה שיתופית של אובייקטים
גישת תורת הקשיחות

דאגלס גולדנברג