2024

148

Rafael Herguedas Gastón

Multirobot manipulation of deformable objects

Director/es

Sagüés Blázquiz, Carlos
López Nicolás, Gonzalo

ISSN 2254-7606

Prensas de la Universidad
Universidad Zaragoza

Universidad de Zaragoza
Servicio de Publicaciones

ISSN 2254-7606

Tesis Doctoral

MULTIROBOT MANIPULATION OF DEFORMABLE OBJECTS

Autor

Rafael Herguedas Gastón

Director/es

Sagüés Blázquiz, Carlos
López Nicolás, Gonzalo

UNIVERSIDAD DE ZARAGOZA
Escuela de Doctorado

Programa de Doctorado en Ingeniería de Sistemas e Informática

2023

Tesis Doctoral

Multirobot manipulation of deformable objects

Autor

Rafael Herguedas Gastón

Directores

Gonzalo López Nicolás
Carlos Sagüés Blázquiz

Escuela de Ingeniería y Arquitectura
2023

PhD Thesis

Multirobot manipulation of deformable objects

Rafael Herguedas Gastón

Supervisors:
Gonzalo López Nicolás
Carlos Sagüés Blázquiz

Universidad Zaragoza

Programa de Doctorado en Ingeniería de Sistemas e Informática

2023

Acknowledgments

Firstly, I would like to express my sincere gratitude to my supervisors Gonzalo López and Carlos Sagüés, for the opportunity they gave me to pursue the PhD and their invaluable guidance throughout these years.

For accepting me as an external collaborator in their research groups, and giving me the opportunity to learn from excellent professionals, I would like to thank Prof. Youcef Mezouar and Dr. Máximo A. Roa.

Besides, I want to show my gratitude to all the partners of project CoMManDIA, whose contributions were an essential source of inspiration and motivation for my work in the first stages.

I would also like to thank my colleagues from the Department of Computer Science and Systems Engineering of the University of Zaragoza, especially the exceptional and wonderful PhD students of the ROPERT research group.

For their continuous and fundamental support, I wish to thank my dear family and extraordinary friends from outside the academia.

Finally, I would like to thank the Government of Aragón and the European Union for the financial support I have received during the PhD.

Resumen

Los objetos deformables son un elemento fundamental en incontables procesos industriales y de otros ámbitos, como el doméstico o el sanitario. La automatización a través de plataformas robóticas de aquellas tareas en las que se manipulan estos objetos resulta interesante no solo desde el punto de vista de la mejora de la precisión y la eficiencia de los procesos, sino también por los beneficios que trae consigo para los trabajadores humanos. Estos beneficios surgen, principalmente, en la realización de tareas repetitivas, que conllevan movimientos poco ergonómicos o en las que se manejan sustancias tóxicas o peligrosas. En este sentido, los robots podrían llevar a cabo estas tareas de forma parcial o total, permitiendo con ello a los operarios la realización de tareas más variadas y seguras.

Esta tesis doctoral tiene como objetivo el proponer, analizar y validar distintos sistemas multiagente para la percepción y el control de objetos deformables. Al considerar múltiples sensores y manipuladores, obtenemos un sistema más eficaz, robusto y versátil que el homólogo de un único elemento. Esto es especialmente cierto en el presente caso, donde el sistema es altamente sub-actuado debido a los infinitos grados de libertad del objeto manipulado.

En primer lugar, analizamos el estado del conocimiento en el ámbito de estudio. Tras identificar los principales trabajos de este campo, los clasificamos de acuerdo a una serie de criterios diferenciadores. A partir de esta clasificación, extraemos conclusiones en cuanto a los aspectos comunes, principales problemas abiertos y líneas de investigación más importantes.

El siguiente aspecto que tratamos es la percepción de objetos deformables con un sistema multicámara. Abordamos esta tarea como un problema de cobertura, en el cual el objetivo se establece en términos de completitud y precisión del contorno percibido. Para definir un objetivo de cobertura factible, estudiamos las propiedades de visibilidad máximas de formas generales desde un enfoque geométrico. A continuación, proponemos dos métodos alternativos basados en optimización para resolver este problema, desde un punto de vista de información global y local. Estos métodos son evaluados y comparados mediante la simulación de distintos casos de aplicación.

Tras analizar la percepción, proseguimos con el cálculo de acciones útiles para manipular objetos deformables. Concretamente, abordamos el transporte de objetos deformables mediante sistemas multirobot. En este caso, proponemos un nuevo modelo de deformación basado en la caja envolvente deformable del objeto manipulado. El modelo, con dinámica de doble integrador y calculado a partir de un conjunto de medidas iniciales, relaciona la evolución temporal en las dimensiones y orientación de la caja envolvente con la acción colectiva de los manipuladores robóticos. Un conjunto de controladores nominales explota el modelo para llevar las dimensiones, orientación y posición del objeto al estado deseado. Dado que el control nominal no tiene en cuenta la corrección del rumbo ante obstáculos, aumentamos el mismo con una nueva función barrera de control que garantiza la evitación de colisiones. Validamos el controlador integrado con la función barrera en simulaciones con obstáculos estáticos y dinámicos.

El sistema anterior posibilita un control del objeto deformable limitado a su caja envolvente. Con el propósito de controlar la configuración del objeto de forma más precisa, presentamos un controlador de formaciones de robots alternativo para resolver el problema de transporte de objetos deformables. Mediante un conjunto de controladores para robots con dinámica de doble integrador, conseguimos controlar la forma, escala, posición y orientación de la formación de manipuladores que sostiene el objeto. En esta propuesta, y a diferencia de la anterior, no modelamos la deformación, sino que asumimos que el objeto es suficientemente flexible y que la formación de robots representa adecuadamente la forma del objeto. Para garantizar que el funcionamiento del controlador es el deseado, estudiamos en profundidad sus propiedades en términos de desacoplamiento de las variables de la formación, estabilidad y convergencia al estado objetivo. Además, proponemos un conjunto de funciones barrera para evitar colisiones y sobreestiramientos en entornos con obstáculos estáticos y dinámicos. Este nuevo controlador es aplicado satisfactoriamente en escenarios tanto simulados como reales.

En las propuestas anteriores, consideramos que los objetos son suficientemente resistentes como para poder ser agarrados utilizando métodos convencionales. En caso de que el objeto manipulado sea fino y frágil, éste podría resultar dañado al ser agarrado con una estrategia de este tipo. Por ello, proponemos un sistema de manipulación secuencial basado en acciones de empuje, sobre un plano en el que descansa el objeto. Mediante tales acciones, podemos controlar la posición y orientación del objeto, al mismo tiempo que contenemos su deformación dentro de un rango de seguridad. Utilizando regresión de procesos Gaussianos y optimización Bayesiana, podemos modelar las propiedades del objeto sujeto al empuje, actualizar el modelo resultante a lo largo del tiempo y calcular acciones efectivas en cada instante. Mostramos la eficacia de esta propuesta en diferentes escenarios de simulación.

En conclusión, las contribuciones principales de esta tesis son las siguientes: hemos desarrollado y validado nuevos métodos de percepción de objetos deformables con sistemas multicámara; hemos propuesto técnicas de modelado de deformación y control multirobot aplicadas a la tarea de transporte de objetos deformables; y hemos estudiado un método novedoso para manipular objetos deformables frágiles y finos, sobre superficies planas.

Abstract

Deformable objects are a paramount element in uncountable processes of industrial and other domains, such as the domestic and the healthcare ones. Automation through robotic platforms of those tasks where deformable objects are manipulated is interesting not only from the perspective of improving the accuracy and efficiency of processes, but also for the benefits this implies for the human workers. These benefits arise, mainly, when performing repetitive tasks, that involve uncomfortable movements or in which toxic or dangerous substances are handled. In this sense, robots could carry out partially or totally such tasks, bringing about the possibility of performing more varied and safe tasks to the workers.

The present PhD thesis aims at proposing, analyzing and validating different multiagent systems for perception and control of deformable objects. By considering multiple sensors and manipulators, we obtain a system that is more effective, robust and versatile than the single-element counterpart. This is particularly true in the present case, where the system is highly underactuated due to the infinite degrees of freedom of the manipulated object.

Firstly, we analyze the state of the art in the field of this research. After the identification of the main works in this area, we classify them according to a selection of differentiating criteria. From this classification, we extract concluding remarks with respect to the common aspects, main open problems and the most important research lines.

Next, we deal with the perception of deformable shapes by a multicamera system. We tackle this task as a coverage problem, in which the objective is set in terms of completion and accuracy of the perceived contour. With the purpose of defining a feasible coverage objective, we study the maximum visibility properties of general shapes from a geometrical perspective. Then, we propose to solve this problem with two alternative optimization-based methods, from the point of view of global and local information. These methods are evaluated and compared by means of simulations of different application cases.

After analyzing perception, we continue with the search of useful actions for manipulating deformable objects. In particular, we tackle the transport of deformable objects by multirobot systems. In this case, we propose a new deformation model based on the deformable bounding box of the manipulated object. The model, with double-integrator dynamics and computed from a set of initial measurements, maps the temporal evolution of the dimensions and orientation of the bounding box with the collective action of the robotic manipulators. A set of nominal controllers exploits the model for driving the dimensions, orientation and position of the object to the desired state. Due to the fact that the nominal controller does not take into account the path correction against incoming obstacles, we augment it with a new control barrier function that guarantees collision avoidance. We validate the integrated controller with the control barrier function in simulations with static and dynamic obstacles.

The previous system enables the control of the bounding box of a deformable object. With the purpose of controlling the configuration of the object more accurately, we present an alternative formation controller for solving the problem of transporting deformable objects. By means of a set of controllers for robots with double-integrator dynamics, we are able to control the shape, scale, position and orientation of the manipulators' formation grasping the object. In this proposal, and in contrast to the former one, we do not model deformation, but we assume that the object is flexible enough and the robotic formation represents well enough the shape of the object. For guaranteeing the desired performance of the controller, we study its properties in depth in terms of uncoupling of the formation variables, stability and convergence to the target state. In addition, we propose a set of control barrier functions for collision and overstretching avoidance in environments with static and dynamic obstacles. This new controller is successfully applied in simulated and real scenarios.

In the previous proposals, we consider the objects are resistant enough for being grasped utilizing conventional methods. In case the manipulated object is thin and fragile, it could be damaged when being grasped with such strategies. Therefore, we propose a sequential manipulation system based on pushing actions, on a plane where the object lies. Through such actions, we can control the position and orientation of the object, at the same time we contain its deformation within a safety threshold. By applying Gaussian process regression and Bayesian optimization, we can model the properties of the object under pushing, update the resulting model over time and compute effective actions at every instant. We show the performance of this proposal in different simulation scenarios.

In conclusion, the main contributions of this thesis are the following: we have developed and validated new perception methods for deformable objects with multicamera systems; we have proposed deformation modelling and multirobot control techniques applied to the task of transporting deformable objects; and we have studied a novel method for manipulating thin and fragile deformable objects, over flat surfaces.

Contents

Chapter 1

Introduction

1.1 Motivation

STRICTLY speaking, all materials with practical interest exhibit some kind of deformation when certain force is applied to them. In fact, almost every industrial product contains parts that show macroscopical flexibility. This implies that not only the production, but also the usage of these products involve manipulation tasks where deformation must be controlled in some manner. Therefore, the importance and interest of automatizing those kinds of tasks is notable in social, industrial and economical terms [56]. Traditionally, the objects manipulated by robotic platforms were considered rigid. This has been likely due to the bounded mechanical properties of some materials in common applications, e.g metals, that require high forces for undergoing deformations. Besides, there are inherent difficulties that arise when dealing with deformation: infinite degrees of freedom of the objects, different deformation modes (traction, compression, torsion, etc.), time-varying mechanical properties and non-fixed contact points, among others. Nowadays, the manipulation of rigid objects is still an active research field with open issues, but there is a growing interest towards including deformation as an additional parameter.

In the field of mechanics of materials, the deformation of a segment (or, more rigorously, its elongation per unit length or strain) is defined as the ratio between the elongation divided by the segment's total length [51]. There are two different types of deformations: elastic and plastic. Elastic deformations correspond to those that vanish when the applied load is released. This behavior is the result of small changes in the interatomic spacing and the stretching of interatomic bonds [24]. Examples of materials showing elastic deformations are most metals, for strains under 0.005, and elastomers. In contrast, plastic deformations remain after releasing the acting force. They correspond to a rupture and subsequent re-forming of bonds after a relative movement of a large number of atoms [24]. E.g., metals for strains over 0.005 and modeling clay exhibit plastic deformations. Yet simple in appearance, this basic notions are only the prelude of the complex physics that govern deformation in the general case. In the framework of this thesis, the goal is not studying these complex mechanisms in detail, but to understand and control them as required for manipulating a selection of paradigmatic deformable objects, that can be found in common applications. For the purpose of robotic manipulation, we deal with deformation as follows: by designing novel coverage strategies for perceiving the contour of deformable objects over time; through new models that predict deformation in real-time applications; with formation control methods that steer the manipulated object to desired configurations; and developing manipulation strategies that maintain deformation of fragile objects within a safe range.

These developments have been motivated mainly by project CoMManDIA, from the Interreg Sudoe european programme [72]. With the objective of enhancing the working conditions of human operators and increasing the productivity and efficiency of industrial processes, CoMManDIA aimed to develop new solutions for robotic manipulation of deformable objects. In the industrial sector, robotic systems can help to reduce the risk of accidents, the fatigue and improve the overall health and safety conditions of human workers by performing tasks that are physically demanding, repetitive and hazardous. By doing so, human operators can focus on tasks that require higher levels of cognitive processing, such as decision-making and problem-solving ones. Furthermore, this allows reducing production defects, adapting the manufacturing process to objects of different shapes and sizes and quickly reconfiguring the work flow to changes in product design or demand. But the positive impact of robotic solutions of this kind is not limited to the industrial field. By properly adapting the manipulation methods, other sectors can obtain benefits in multiple areas. In this sense, robots could deal with domestic tasks like cloth folding and cloth spreading, thus assisting people in tedious efforts. Healthcare is another field that could implement robotic-based systems, for procedures such as accurately manipulating tissues in surgical interventions and the transport of patients. Overall, robotic solutions for manipulating deformable objects may lead to changes in the nature of work for human operators in a broad variety of contexts.

Another key aspect of the thesis is that it is focused on multiagent systems. Considering multiagent strategies for perceiving and manipulating deformable objects makes sense due to the advantages they present versus the single-agent ones. Some of these advantages are the possibility of dealing with highly complex tasks, the distributed nature of the system, a smaller number of technical requirements in terms of capabilities of the units, higher robustness and adaptability and higher scalability and flexibility [48].

1.2 Objectives

As stated in the previous section, the main purpose of this thesis is to study new multiagent systems for manipulating deformable objects in general-purpose applications. The first step to tackle is the thorough study of the field of multirobot manipulation of deformable objects. In this still emerging but rapidly growing area, such study would provide the knowledge foundations for subsequent developments. A series of classification criteria should be proposed and applied to analyze the main trends in deformation modeling, manipulated objects, control strategies and perception methods, among other characteristics. Then, the goal can be decomposed in a series of objectives with specific sub-goals:

1. **Perception of deformation**. Before considering the manipulation task itself, it is needed to find new flexible, versatile and effective procedures for measuring object deformation. Flexibility and versatility are required for the compatibility of the perception system with the control one, as perception will be a key element of the subsequent deformation control approaches. In turn, the proposal should guarantee that deformation is measured with enough accuracy and completeness. This will serve as an indicator of the performance and quality of the manipulation process.

2. **Transport with deformation control**. With the purpose of driving an object grasped by multiple robots to a target configuration (i.e. position, orientation and dimensions), without collisions, novel solutions in terms of modelling and control should be investigated. The information given by the measured deformation can be exploited to improve the accuracy of the model, and to evaluate the performance of the system. Controlling deformation will be of paramount importance for not damaging the object and achieving the task goals.

3. **Non-prehensile deformation control**. In case the manipulated object cannot be grasped with standard methods, due to its fragility, alternative non-prehensile manipulation strategies must be studied. For this kind of task, it is assumed that the object lies on a supporting medium, and the goal is steering it to a target configuration by means of multiple pushing actions. Such scenario requires inferring not only the structural properties of the object under pushing, but also the effects over deformation of the friction between the object and the supporting surface.

1.3 Outline

The content of the present thesis is structured in seven chapters, that begin in **Chapter 1** with the introductory sections and end with the summary, closing remarks and tentative future research in **Chapter 7**. In the remaining five thematic chapters, the different research blocks are developed as follows:

- **Chapter 2** reports the study of the state of the art in multirobot manipulation of deformable objects. The main approaches, at the time this thesis started, are identified and classified according to different criteria: modelling strategy, dimensionality of the manipulated object, control method, perception system, type of manipulation actions, practical application, and manipulation platform. Additionally, a selection of multirobot control-related studies, which do not explicitly tackle manipulation of deformable objects, is listed and described for their potential utility in this kind of tasks. From this analysis, the active research lines, leading strategies and open issues are identified and summarized.

- In **Chapter 3** a multicamera system is proposed for coverage over time of deformable shapes. The perception task is defined in terms of visibility and resolution of the shape's contour, parameters that quantify the completion and accuracy of the coverage, respectively. A method for obtaining the maximum visibility and resolution limits of the shape at a specific time instant, required for defining a feasible coverage objective, is described. Afterwards, configurations that achieve the target coverage with minimal sensors, over the expected deformation process, are obtained in an offline optimization framework. Alternatively, an online variant is presented for fast updating the system in real-time perception tasks. These methods are validated through different test cases, in simulated and real-world setups.

- **Chapter 4** shows a multirobot system for transport of deformable objects in 3D environments. With the goal of translating the object without collisions to a target position, a novel Deformable Bounding Box model is proposed. This model allows predicting the dimensions and orientation of the bounding box that contains the object, under collective manipulation actions applied by the robots. The manipulation actions are regulated by a set of nominal controllers for robots with double-integrator dynamics, which deform, rotate and translate the object as required in the task. Due to the fact that these controllers do not consider collision avoidance explicitly, a Control Barrier Function (CBF) is defined to act when the obstacles intersect with the nominal trajectories. In the framework of a Quadratic Programming-based (QP) controller, the control action is ultimately computed as the closest one to the nominal control action that satisfies the constraints imposed by the CBF. Through simulated experiments, the performance of the method is illustrated in different test cases.

- A different solution to the problem of transporting deformable objects with a team of robots is presented in **Chapter 5**. The method applies formation control in 2D with a set of terms that steer the object to a target configuration, defined as the combination of shape, scale, centroid position and rotation of the formation. A rigorous formal analysis of the controller proves the uncoupling between the formation parameters, as well as the stability of the system and the convergence to the desired state. Similarly to the method of the previous chapter, the robots are modelled with double integrator dynamics and collision avoidance is not explicitly accounted by the formation controller. This is the reason why a set of new CBFs is proposed to avoid collisions with static and dynamic obstacles. Given that the collision avoidance maneuvers could produce undesired deformations, an additional CBF is implemented to avoid overstretching. Again, the final control input is given by the QP as the closest one to the formation control action that satisfies the constraints imposed by the CBFs. In contrast to the previous method, the object deformation is not modelled, and it is assumed that the object is flexible enough and its deformation states are bounded. A set of experiments in simulated and real-world scenarios show the performance of the method in various cases.

- Finally, the interest is directed towards thin fragile objects in **Chapter 6**. For the problem of manipulating this kind of objects to a target position and orientation, while their deformation remains bounded, a novel manipulation strategy is described. This strategy is based on applying a sequence of planar pushing actions on the edges of the object, which lies on a flat supporting medium. With such procedure, it is possible to infer through Gaussian process regression a probabilistic model that maps the pushing actions with the state of the object over time, in terms of deformation and position of the object's centroid on the surface. Simultaneously, an Adaptive Bayesian Optimization procedure determines pushing actions that are most likely going to decrease the task errors. The solution space where to find the optimal pushes is restricted by a series of adaptive constraints, which aim at accelerating the exploration process and obtaining effective control actions. These developments are tested and verified in a realistic simulation environment with the Pybullet simulator.

1.4 Contributions and publications

The research works that integrate this thesis have been developed at the Engineering Research Institute of Aragon (I3A), University of Zaragoza, in the Department of Computer Science and Systems Engineering (DIIS). Additionally, some of them are part of two 3-month international collaborations with the *Institut Pascal*, *Université Clermont Auvergne* (France), and the Institute of Robotics and Mechatronics, DLR German Aerospace Center (Germany). These works are supported by a series of scientific publications in international journals and conferences. In this section, the thesis contributions are reported jointly with the corresponding references where they appear.

The identification and further analysis of the relevant works in the field of multirobot manipulation of deformable objects represents the first survey about this specific topic, and includes a novel classification that highlights differentiating properties of the methods. Besides, the discussion and comparison of those properties among the analyzed studies generate new insight about the current perspectives in this area of knowledge. The publication associated to these contributions [58] is:

1. R. Herguedas, G. Lopez-Nicolas, R. Aragues, and C. Sagues, "Survey on multi-robot manipulation of deformable objects," in *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sep. 2019, pp. 977–984. DOI: 10.1109/ETFA.2019.8868987.

Multicamera setup for perception of deformable shapes. For the purpose of perceiving a deformable 2D shape over time, we analyze the visibility properties from a novel geometrical perspective. As a result of this analysis, we propose a new technique that allows computing the maximum visibility and resolution of the contour at a specific distance. Then, we describe a novel formulation of the coverage task as a continuous-space optimization problem, and we obtain two alternative solutions. The first solution provides a minimal set of cameras for covering a target contour, over a prescribed deformation guideline and according to the coverage objectives. In case the deformation process is unknown, we present a different technique where the configuration of the cameras is quickly updated through local optimizations. These contributions are featured in a national conference paper [60], two international conference papers [59], [61] and a journal paper [63]:

2. R. Herguedas, G. Lopez-Nicolas and C. Sagues, "Sistema multi-cámara mínimo para percepción de formas deformables," in *VIII Jornada De Jóvenes Investigadores del I3A*, May 2019, DOI: 10.26754/jji-i3a.003578.
3. R. Herguedas, G. Lopez-Nicolas and C. Sagues, "Multi-camera coverage of deformable contour shapes," in *IEEE International Conference on Automation Science and Engineering (CASE)*, Aug. 2019, pp. 1597-1602, DOI: 10.1109/COASE.2019.8843053.
4. R. Herguedas, G. Lopez-Nicolas and C. Sagues, "Experimental multi-camera setup for

perception of dynamic objects," in *Workshop ROMADO, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2020, pp. 11874-11878, ISBN: 978-1-7281-6211-9.

5. R. Herguedas, G. Lopez-Nicolas and C. Sagues, "Coverage of deformable contour shapes with minimal multi-camera system," *Measurement*, vol. 190, 2022, pp. 110693, DOI: 10.1016/j.measurement.2021.110693.

Multirobot system for transport of deformable objects with DBB model. We formulate a new Deformable Bounding Box (DBB) model for predicting the 3D shape and orientation around the vertical axis of the bounding box of an object, under the collective action of a team of manipulators. The DBB model is then exploited by a QP-based controller, which steers the manipulated object to a specific configuration (position, orientation and shape) thanks to a set of nominal controllers for manipulators with double-integrator dynamics. In this optimization framework, collision and overstretching avoidance are guaranteed with novel constraints based on CBFs and instantaneous differentiation. Two different research works [62], [64] show the previous contributions:

6. R. Herguedas, G. Lopez-Nicolas and C. Sagues, "Collision-free transport of 2D deformable objects," in *21st International Conference on Control, Automation and Systems (ICCAS)*, Oct. 2021, pp. 430-435, DOI: 10.23919/ICCAS52745.2021.9650027.
7. R. Herguedas, G. Lopez-Nicolas and C. Sagues, "Multirobot Transport of Deformable Objects With Collision Avoidance," *IEEE Systems Journal*, vol. 17, no. 2, pp. 3224-3234, DOI: 10.1109/JSYST.2022.3213972.

Formation control in transport tasks of deformable objects. The difficulties associated to the deformation modelling are bypassed in a different approach, that leverages formation control for steering a team of robots to a specific configuration. In this case, we assume that the object grasped by the robots is flexible enough to follow the movement of the formation. With a novel formulation for robots with double-integrator dynamics, which includes different terms for controlling the shape, scale, position and orientation of the formation, the system achieves the desired configuration with full uncoupling of the formation parameters. The properties of the controller in terms of uncoupling, stability and convergence are comprehensively analyzed. Then, we augment the formation controller with a set of new object-to-obstacle and overstretching avoidance CBFs in a QP-based framework. As an alternative for mobile manipulators with nonholonomic motion constraints, we also propose a control method that allows transporting and simultaneously deforming an object to a goal shape. We have published the former contributions in two international conferences [57], [91] and are under review in a journal paper:

8. G. Lopez-Nicolas, R. Herguedas, M. Aranda and Y. Mezouar, "Simultaneous shape control and transport with multiple robots," in *IEEE International Conference on Robotic Computing (IRC)*, Nov. 2022, pp. 218-225, DOI: 10.1109/IRC.2020.00042.

9. R. Herguedas, M. Aranda, G. Lopez-Nicolas, C. Sagues and Y. Mezouar, "Multirobot control with double-integrator dynamics and control barrier functions for deformable object transport," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2022, pp. 1485-1491, DOI: 10.1109/ICRA46639.2022.9812378.
10. R. Herguedas, M. Aranda, G. Lopez-Nicolas, C. Sagues and Y. Mezouar, "Double-integrator multirobot control with uncoupled dynamics for transport of deformable objects," accepted in *IEEE Robotics and Automation Letters*, 2023.

Manipulation of thin fragile deformable objects. By means of a multirobot system and a new technique based on sequential pushing and Adaptive Bayesian Optimization, it is possible to manipulate thin fragile deformable objects to a specific position and orientation in 2D. At the same time, the object deformation is controlled to remain within an admissible range. The time-varying physics of deformation and friction are learned through Gaussian Process Regression with information from the pushing actions. Then, we combine the resulting model with new adaptive constraints that accelerate the process of finding effective actions at every time step. A conference paper that develops these contributions is under preparation at the present time:

11. R. Herguedas, A. M. Sundaram, G. Lopez-Nicolas, M. A. Roa and C. Sagues, "Adaptive Bayesian optimization for robotic pushing of thin fragile deformable objects," under preparation.

Chapter 2

State of the art in multirobot manipulation of deformable objects

In this chapter, we analyze relevant state-of-the-art approaches concerning manipulation of deformable objects by multiple robots. Then, we classify them according to different specific criteria. As a result of the analysis and classification, we identify the main trends, challenges and most important research directions.

2.1 Introduction

In the field of robotics, as previously mentioned, manipulation of deformable objects is still an open problem. As opposed to the well-studied framework of rigid objects manipulation, being able to predict how the object is going to behave under the effects of a certain manipulation action is a crucial and challenging aspect when dealing with deformable materials. Production of clothes and footwear, food handling, toys manufacturing and surgery are some of the applications involving these kinds of objects. One of the main interests of automatizing some of these sector's manipulation tasks is to reduce the health hazards for human workers, who have to go through uncomfortable, unpleasant and even dangerous works.

When dealing with certain types of deformable objects one can find that they are too big, too heavy, difficult to grasp, too fragile or too soft for being manipulated by a single robot. Therefore, in order to improve the performance of the robotic systems in terms of accuracy, computational cost and flexibility, multiple robotic manipulators with the same or different roles must be considered to carry out the task. Previous surveys on the topic of autonomous manipulation of deformable objects have been developed in the recent years [79], [76], and in particular the reader is referred to the comprehensive survey performed by Sanchez *et al.* [111] for a global understanding of the current state of the art in robotic manipulation of deformable objects. However, none of them is focused specifically on manipulation of deformable objects by multi-robot teams. We have studied, and subsequently classified according to multiple criteria, recent approaches which are relevant in the field of manipulation of deformable objects by multiple robots. The classification encompasses the following elements: the deformation model they consider, the dimensionality of the deformable object, the control strategy they follow, the perception system they apply, the predominant manipulation actions, the practical application they tackle and the type of robotic manipulators. Besides, a selection of multirobot control systems is analyzed, for their potential utility in manipulation tasks of deformable objects. After the analysis and classification, we summarize the leading strategies, the common issues and the future research alternatives. We present Table 2.1 at the end of the chapter as a summary of the surveyed and classified works of the state of the art. It is worth mentioning that we consider dual-arm robots as a multi-robot manipulation system, due to the fact that each arm represents an independent manipulation unit.

2.2 Classification according to the model

Due to the highly-dynamic behavior of deformable objects, correctly modelling deformation is one of the main concerns when handling these kinds of objects. In the past, many studies that tackled manipulation of deformable objects were based on precomputed models, but nowadays there is a clear tendency on exploiting methods which learn an online deformation model. Moreover, there are others that do not consider any deformation model of the object. This model-free perspective improves both robustness and generality of the methods. It is important to remark that the following discussion only includes methods that are implemented inside the manipulation algorithms. It does not consider models that represent the deformable object in simulation tests.

2.2.1 Methods using precomputed models

Those approaches where a deformation model is computed offline, previously to the system working in real time, are analyzed and compared in this section.

Mesh models, with either discrete (mass-spring-damper elements) or continuous (finite element method) formulation, are common candidates to represent the deformable object. Das and Sarkar [35] consider the problem of handling a 2D deformable object exhibiting rheological deformations (elasticity and viscosity properties) with a group of robotic manipulators, or a robotic hand with multiple fingers. They model the deformable object by means of a mass-spring-damper elements mesh, and deformation is controlled by applying an optimization technique over the mesh boundary nodes. Also Li *et al.* [86] utilize mesh models for the purpose of manipulating deformable objects (clothes). They have built a precomputed database which contains 3D mesh models of different kinds of garments, all of them simulated under the effects of gravity and picked up in multiple poses. When a piece of clothing is grasped for performing a manipulation task, the recognition system creates its 3D mesh model and extracts some relevant features to be compared with the ones of the precomputed models from the database. After finding a correspondence, the algorithm obtains the optimal manipulation trajectories leading to the desired state. Similarly to the previous approach, Jia *et al.* [74] follow a strategy that is based on building a visual dictionary in an offline way and using it afterwards to manipulate highly deformable materials. Their dictionary consists of a set of vectors that store visual feedback data and end-effector velocities, whose mapping is obtained by means of a training phase. Different goal configurations are computed at runtime depending on the selected manipulation task, and using sparse linear representation the velocity of the controller is computed from the visual dictionary. Manipulation tasks like folding and flattening of cloth pieces with varying material properties are performed in experimental tests, with a dual-arm robot and the dual-arm robot in collaboration with a human. Higher accuracy values may be obtained from finite element models in comparison with the mass-spring-damper mesh models. Duenser *et al.* exploit these kinds of models in [42] for performing user-specified deformations over elastic parts by means of a dual-arm robot, under the quasi-static system assumption. Their approach performs an efficient real-time optimization in which the Jacobian relating the joint angles' changes to the variations of the object's shape is continuously obtained. A different method is presented by Long *et al.* [89] for obtaining the direct and inverse dynamic models of a group of two manipulators cooperatively carrying a flexible object. This technique can be applied to rigid, articulated and flexible objects whose deformation can be expressed with a normal distribution. While the object is modeled using the generalized Newton-Euler formalism, the robotic manipulators are modeled with the rigid arm equations and the kinematic Jacobians, and the two subsystems are linked together by the wrench applied at the end effector's grasp.

The main disadvantage of the aforementioned methods is that the model has to be modified and recomputed each time the properties of the deformable object change, e.g., a T-shirt of a different size, or different material, etc.

2.2.2 Methods using learned models

Instead of using a model with fixed parameters, the following approaches focus on learning the deformation model parameters in an online manner.

With the aim of deforming 3D rheological objects, Higashimori *et al.* [65] propose a two-phase strategy in which the parameters defining the behavior of the object model are obtained previously to performing the desired deformation action. This approach is able to deform 3D objects to a desired state but only in one direction, due to the fact that it relies on a 1D four-element model (2 springs and 2 dampers) of the object. A more recent method in which a local deformation model is approximated online, through an estimation-recalibration algorithm, is proposed by Navarro-Alarcon and Liu [100] for the purpose of deforming 3D soft objects into 2D desired contours. In this approach, one active and one or more passive grippers that follow translational quasi-static movements are servo-controlled to deform the object. The object's 2D shape is represented with Fourier series, and its physical properties are completely unknown. Also in [102] Navarro-Alarcon *et al.* describe a method for deforming 3D elastic objects with two robotic arms and achieve different positions and shapes by estimating the Jacobian of the deformable object. Hu *et al.* [69] present a similar method with improved properties in terms of convergence and dynamic behavior. They estimate the object's deformation by means of an algorithm called FO-GPR (Fast Online Gaussian Process Regression), which obtains a nonlinear deformation function and updates it at each time step. In contrast to the standard GPR methods, the FO-GPR removes uninformative observation data, which allows to substantially decrease the computational cost of the algorithm and also improves the accuracy of the model. Multi-robot low-level manipulation tasks, in which the deformation function does not vary at different stages (rolled towel bending, towel folding, etc.), are successfully performed with this technique in experimental tests, even in the presence of partial occlusions. A different Jacobian-based strategy is developed by Berenson [20] in order to perform locally defined quasi-static manipulation tasks of 1D and 2D deformable objects, with a single or multiple robotic manipulators. His algorithm exploits the concept of diminishing rigidity (from the grasping point to the rest of the object) to compute an approximate Jacobian of the object, which is corrected afterwards to include excessive stretching constraints. This approach depends on manual tuning for setting an adequate evolution of rigidity, and it cannot represent properly the properties of heterogeneous deformable objects. However, a recent study by McConachie and Berenson [94] that is based on the MAB concept (Multi-Armed Bandit) solves the problem of automatically selecting an appropriate deformation model. In particular, their algorithm is called KF-MANDB, and extends the standard MAB technique to consider a nonstationary, inter-dependent, Kalman-filtered framework. They consider a set of grippers to perform a specific deformable object manipulation task, and they build a model database with approximate Jacobian models tuned with different parameter values. At the same time that the manipulation task is performed, the algorithm evaluates the utility of each model (i.e. how accurate each model is for representing the deformation state) and selects the one with the highest value.

Some additional methods that consider learning a deformation model are [84], in which the stiffness matrix of a finite element model is obtained by probing the material, and [142], in which the object's shape is approximated by Fourier series. Also a combined deformation-projection Jacobian is estimated online in [2].

2.2.3 Model-free methods

Even more flexibility can be attained, *a priori*, if the behavior of the soft object is not related to a specific deformation model. Next methods take advantage of this and develop model-free multi-robot techniques for manipulating deformable objects.

Indirect Simultaneous Positioning is a concept studied by Wada *et al.* in [130] with the goal of controlling the position of a set of points lying within the contour of a 2D deformable shape. They classify the interest points of the shape into manipulation points, to be grasped by the grippers, and positioned points, whose position is to be controlled. Then, they show two different PID control methods for achieving the control objective: a first one that relies on an approximate deformation model, and a second one for small deformations in which the deformation model is not needed. Deformations are here restricted to the plane of the 2D object. Another model-free approach proposed by Bai and Wen [16] deals with the problem of flexible load transport from a formation control perspective. They have developed a decentralized control method for the collaborative manipulation of a deformable object by a group of robotic manipulators. Despite the fact that the geometry of the object and the position of the grasping points have to be known in advance, no model of the object is considered. Some assumptions made in this study are that the deformation of the object is small and only appears in a small area around the grasping points, and also that only translational movements are performed. Within the field of robotic cutting of deformable objects, Long *et al.* [88] follow a *pressing and slicing* strategy for separating soft parts without explicitly modeling the object. Apart from the cutting robot, which is equipped with the cutting tool, a second robot is considered to provide a pulling force (whose magnitude is obtained from experiments) that reduces the necessary cutting force.

2.3 Classification according to the manipulated object

A different classification of the manipulation strategies can be performed from the perspective of the manipulated object. Real deformable objects are always lying on the 3D space, but some of them are studied by omitting those dimensions which are of a much smaller magnitude than the rest. For instance, a rope can be studied by considering it a 1D entity with null cross section provided that its length is much larger than its thickness in both transversal directions. Thus, the following subsections differentiate the multi-robot manipulation approaches by the dimensionality of the deformable object.

2.3.1 Unidimensional objects

As reported before, ropes and also cables are typical examples of unidimensional objects. A recent approach by Zhu *et al.* [142] tackles the problem of manipulating 1D flexible cables to match a desired 2D contour, by means of a dual-arm robot equipped with special-purpose grippers. Inspired by [100], they represent the cable's shape with truncated Fourier series. To reach a compromise solution between accuracy, computational cost and under-actuation, only two harmonics of the series are accounted. As opposed to [100] this method considers also rotational movements, but it is not capable of predicting whether a final shape is reachable or not. The challenging task of in-air knotting of 1D ropes is tackled by Kudoh *et al.* in [82]. After extracting a set of hand motions with high reusability (*skill motions*), they develop a specialized hardware system to perform the in-air knotting task. In particular, this system consists of a dual-arm robot equipped with three-finger hands and an RGB-D camera, which is utilized to achieve the initial grasping of the rope. Also in [125], [126] a 1D rope is deformed to a target configuration by two robotic arms. The algorithm created by Tang *et al.* in [125] is called TSM-RPM (Tangent Space Mapping-Robust Point Matching), and maps the evolution of the object's curve tangents to reach the desired configuration between different initial configurations, by means of a non-rigid transformation function. The authors show in simulations how the TSM-RPM algorithm outperforms the TPS-RPM (Thin Plate Spline-Robust Point Matching) in terms of overstretching avoidance and fidelity of the final configuration with respect to the desired one.

2.3.2 Bidimensional objects

The main examples of 2D deformable objects, in terms of the number of approaches considering them, are the cloth-like ones. This type of object is unable to withstand compression forces, and usually shows high rigidity values when submitted to traction forces. Thin panels of deformable materials can be considered as 2D objects too.

Alonso-Mora *et al.* [5] propose a hybrid centralized/distributed algorithm for the transport of deformable objects by multiple robotic manipulators. While the centralized approach is considered in order to provide a global guidance to the manipulators, the distributed one enables the manipulators to move according to the global planning without explicit communication between them. This algorithm includes collision avoidance with both static and dynamic obstacles and shape preservation constraints. Three different 2D deformable objects (a foam mat, a bed sheet and a towel) are collaboratively carried to a desired position by a multi-robot team in experimental tests. Deformation is controlled here but only for the purpose of maintaining the structural integrity of the object during the transportation. Focusing also on 2D deformable objects collaborative manipulation, Langsfeld *et al.* [84] develop a multi-robot system that allows to clean plastic parts with two fixed redundant robotic arms of differentiated roles: the first arm cleans the part without deforming or breaking it while the second one holds the object in a proper position. The object is assumed elastic and is modeled with 1D finite elements whose elastic properties are updated as the cleaning task proceeds. Regrasping actions are optimized in order to minimize the part deformation and the cleaning time. A different collaborative situation in which a 2D deformable sheet is manipulated between a person (uncontrollable agent) and a dual-arm robot (controllable agent), equipped with a Kinect camera, is studied by Kruse *et al.* [80]. Initially, the opposite ends of a piece of fabric are grasped by the human and the robot, and the control goal is to minimize the amount of wrinkles and local deformations produced when the human manipulator performs local movements, that distort the initial undeformed state. Collaborative human-robot tasks, besides robot-robot ones, are considered too by Jia *et al.* in [74] for cloth folding, cloth flattening and cloth placing. The effectiveness of this method is affected by some limitations in terms of illumination and relative colors of the clothes in the 2D images.

2.3.3 Tridimensional objects

Being probably the most challenging examples of deformable objects for robotic manipulation, 3D objects range from soft foam pieces to food dough, and represent the most general case in which all spatial dimensions are accounted.

The problem of deforming 3D elastic foam parts to a desired state by multiple robots is tackled by Navarro-Alarcon *et al.* in [101]. They propose a robust vision-based controller which accounts for noise and uncertainty in the model estimation. The elastic properties of the deformable object are unknown, and therefore the Jacobian matrix is estimated online with the Broyden update rule [73]. They perform various experimental tests with one and two active manipulators. 3D soft foam parts are also considered by Long *et al.* for a cutting process in [88]. In this approach, experimental tests are carried out over the soft foam parts, that are cut to a predefined cutting depth through sequential cutting actions. The cutting trajectory is specified by a series of visual markers attached to the object. In [16] a lightweight 3D soccer ball is collaborativelly manipulated by two fixed robotic arms, once fed by a human agent. The manipulation task is divided in two steps: the first one implies statically holding and squeezing the ball, and during the second one, linear or circular trajectories are performed. All the previous approaches consider homogeneous objects whose material properties do not vary across their volume. In [2], however, Alambeigi *et al.* propose a multi-robot method for manipulating 3D compliant objects that show heterogeneous material properties. An online estimation system, based on the Secant approximation and the Broyden's method, obtains the combined deformation-projection Jacobian which allows to predict in real time the deformation and the camera parameters. Thus, their method is able to work with uncalibrated vision sensors. With regard to the control algorithm, a constrained optimization problem is solved with the previously computed Jacobian to accomplish the predefined tasks in an environment with potential disturbances.

2.4 Classification according to the control method

In contrast with the previous classifications, here the focus is on the control aspects rather than in the modeling. The approaches that tackle multi-robot manipulation of deformable objects depend usually on singular control laws and complex algorithms. This makes the control-based classification heterogeneous, since a broad variety of approaches is available, and in some cases the approaches cannot be directly assigned to an specific group due to its uniqueness. However, we propose three different control groups in which several studied methods are suitable for being included. Additional groups that can be considered include strategies based on optimal, nonlinear or Jacobian-based control techniques too, but we only include this information in Table 2.1 for brevity.

2.4.1 Classic control

As long as an approach follows at some level either a proportional or a PID control law, among others, it can be treated as a classical control method. The most recent strategies tend to relegate the classic control techniques to the low-level software layers.

A PD-position feedback controller with gravity compensation is adopted by Sun *et al.* in [122]. This approach considers a general flexible payload whose position and orientation must be controlled by multiple robots, at the same time that vibrations at each contact are suppressed. Deformation of the object is obtained from a finite element model, whose dynamics are decomposed into rigid and flexible components that represent the original undeformed shape and the change in shape due to deformation, respectively. Also Wada *et al.* develop the core of their control strategy in [130] over a classic controller, that in this case is a simple PID control law. They propose two different methods: a model-free PID control system, which is valid only in the domain of small deformations, and a model-dependent PID controller, that provides zero error convergence when the deformation is large. In the latter method a spring mesh approximate model of the object is considered. As opposed to the previous approaches, a classic control method can be found in [5] but at the low-level horizon. Here, velocities of the individual manipulators are controlled by means of a PID control law. This controller may introduce some additional errors in the system when dealing with some types of materials due to a buildup in the integral term. It is important to remark that the use of the classic control law is secondary in this approach, and the main part of the control algorithm (high-level control) is based on advanced planning strategies and convex optimization techniques.

2.4.2 Robust control

In robust control techniques, modeling errors and uncertainties are taken into account with the aim of extending the controller's validity. This type of control strategy is well suited when dealing with deformation models due to the fact that uncertainties are always present in the model parameters.

A robust control strategy is considered in [35], where each manipulator's motion is driven by an independent robust controller that is able to work in the presence of model parameters' uncertainties. The global motion planner makes unnecessary any communication between the system agents. Hu and Vukovich develop a shape control system in [68] derived also from the robust control theory. This method aims to produce a desired out-of-plane deformation on a flexible plate with embedded microactuators and sensors, which are represented as a whole in an integrated mathematical model obtained from the Hamilton's principle. In [101] the authors propose a robust passivity-based controller that has into account the presence of a *time-varying* disturbance in the deformation flow estimation, and in [110] the LMI (linear matrix inequality) optimization allows to identify the dynamic parameters of the robotic structure and to define a robust control strategy. In spite of not being a formally developed robust controller, the method proposed by Hu *et al.* in [69] emphasizes demonstrating the robustness of several aspects it covers. One of these aspects is the selection of the state features, in which task-relevant prior knowledge improves the robustness and effectiveness of the control process. Robustness to moderate levels of occlusion, provided that no significant or fast changes happen in the scene, is also achieved thanks to an online learning mechanism.

2.4.3 Adaptive control

Uncertainties and errors in the model parameters are also assumed in adaptive control methods, and their values are allowed to change over time in order to adapt to the time-varying systems. Learned deformation model methods are natural candidates for the adaptive control strategies.

An example of the last statement is the adaptive control system by Navarro-Alarcon *et al.* in [102] to estimate the object's deformation parameters. For the arms to move in a coordinated way, a saturated velocity controller is developed here. In the context of working without deformation model, Bai and Wen propose in [16] two different control schemes: a scheme where the robots velocity is predesignated and an adaptive control technique in which the group velocity is estimated by each agent. A special case of the latter technique, in which the group velocity is known by a single agent and the rest have to estimate it, is explained and validated in experimental tests too. However, an adaptive control system can also be adopted when a precomputed deformation model is present. Based on the Potential Field Method, that creates an attractive force to the goal configuration and a repulsive one around obstacles, the approach by Dang *et al.* [34] is focused on controlling the shape of a flexible surface. The surface is modeled as a mass-spring-damper mesh, and a group of embedded microactuators is considered to deform it. These actuators are divided into two different groups: absolutely actuated points, in which information about desired point coordinates is provided, and relatively actuated points, where relative distances to other neighboring points are set. Different dynamic shape morphing adaptive control laws are designed for each group of microactuators including parameters uncertainty.

2.5 Classification according to the perception system

As it can be inferred from the previous section, many different control algorithms are found when analyzing the existing approaches in multi-robot manipulation of deformable objects. In turn, each control method depends on different perception systems. This fact motivates the perception-based classification that we present here, where three main groups are identified in terms of the measured data.

2.5.1 Force-based perception

Perception systems focused on forces rely on the fact that interacting with deformable objects necessarily implies a force exchange between all involved agents. For instance, in order to grasp and raise an elastic foam part with a robotic gripper an initial grasping force is required to prevent slipping during the lifting action. Afterwards, a second vertical force must be applied to compensate the part's weight and raise the part. By measuring and controlling these forces, some deformable objects manipulation tasks can be successfully carried out.

This fact is shown in [16], where a force perception system is designed. Their decentralized control method considers the contact forces between the robotic manipulators and the flexible load in order to describe the deformation of the soft object, and also to provide an implicit way of communication to the manipulators. These contact forces are maintained by the controller to avoid sliding of the object during the manipulation task. Delgado *et al.* also consider contact forces by means of tactile sensors in [37], in order to develop an agile and adaptable model-independent multi-robot system for dual-arm in-hand manipulation tasks. They propose a novel representation of the tactile data based on tactile images, which are obtained through a combination of dynamic Gaussians. This representation allows them to design a manipulation controller that maintains and adapts the contact configuration according to the task requirements.

2.5.2 Vision-based perception

As the manipulation process evolves, deformation appears and the shape of the object changes. By monitoring these changes with a vision system, the relation between the motion of the manipulators and the deformable object may be obtained, and afterwards the derived model can be utilized by the control algorithm to produce the desired deformation.

Clear examples of vision-based perception systems are included in most of the methods developed by Navarro-Alarcon concerning multi-robot manipulation of deformable objects, like [101], where an energy-based dynamic-state feedback velocity controller is developed. In this approach, deformation of the manipulated object is tracked by a visual feedback system, in which the feedback points are treated with a nonlinear function to constitute a deformation feature vector. Four different types of deformation are defined for the control purposes: point-based, distance-based, angle-based and curvature-based. Calibration of the vision system is not needed here. In [102] the positions of multiple visual markers, which are placed over the surface of the deformable object, are measured with the camera in order to obtain the position and shape errors. This approach can cope with uncalibrated kinematic transformations too. Again, the Fourier-based controller in [100] constantly updates and recalibrates, if necessary, the local deformation model using the vision sensor data. This perceived data also include the full contour of the deformable part, that allows to compute the shape error. One requirement of this method is that a high contrast is needed between the manipulated object and the image background. The recognition system by Li *et al.* in [86] is based on the Kinect sensor. They perform a preliminary 3D segmentation, which obtains the masks of the garments on the depth images, that is followed by the KinectFusion method invocation, which provides the 3D reconstruction. Binary features are extracted from the 3D reconstructed model in order to make the comparison with the models from the database. The Kinect sensor is also utilized by Tang *et al.* in [126] to obtain point clouds of a rope lying on a flat surface. With the aim of manipulating the rope to get some desired configurations, they have designed a multi-robot system based on the CPD (Coherent Point Cloud) non-rigid registration method, which obtains a smooth transformation function from two different point clouds. This method shows strong robustness under occlusions and allows to perform the next three sequential steps: state estimation, task planning and trajectory planning.

2.5.3 Force and vision-based perception

By combining the advantages of force and vision perception systems, a more robust and accurate control method may be obtained. In some cases in which a specific tensional state must be induced in the object for the force control to work, the vision system is the one in charge of driving the object to that tensional state. This happens in [80], where a hybrid force-vision controller is proposed. The vision system must drive the manipulated 2D sheet to a taut state, so that the force controller can start to decrease the amount of wrinkles (the system is unable to measure force while the sheet is not taut), by either moving or applying traction to the sheet to counteract the deformations created by the human manipulator. Also in [88] an adaptive force-vision control system is utilized to separate deformable objects. Here, the force controller prevents global deformation or damage in the area around the cut, and the vision system updates online the trajectory according to the sensed deformation and the modeling errors. A different strategy is followed in [5] with a low-level velocity controller that takes into account the forces that are transmitted through the manipulated object. Despite the fact that these forces cannot be sensed (i.e. they are virtual forces), they act in the decentralized planner as an indirect communication channel which is complemented with an inter-agent vision system. This vision system is implemented on each robotic manipulator, and obtains the position and velocity of the neighboring agents. The force controller in [65], which regulates the loading over the deformable object and obtains feedback data from a load cell, is also complemented with a vision system that monitors the object deformation with a camera.

2.6 Classification according to the manipulation actions

Manipulation of deformable objects involves a sequence of different individual actions that must be performed by the robotic system. These low-level basic actions can be classified into two groups: deformation actions, which consist in inducing relative displacements of the deformable object, and transport actions, which produce absolute displacements of the centre of gravity of the object. Thus, this classification differentiates between methods where the main contribution is provided either by deformation or by transport actions.

2.6.1 Deformation actions

Predicting and controlling deformation is usually the foremost concern when manipulating deformable objects due to these reasons: manipulation tasks often require the object to be deformed in an specific way, and also the “unstable” behavior they present may result in damaging the manipulators or the objects themselves in case the deformation is not controlled.

An interesting study about deformation actions can be found in [28]. Cherubini *et al.* propose a vision-based method for deforming 3D materials exhibiting plastic deformation that includes the following initial assumptions: a) it exists a finite set of deformation action types (pushing, tapping and incising) and b) the deformation actions have a local influence into the object. A preliminary study with human participants is performed in order to validate these statements, in which the participants are requested to form a certain shape with kinetic sand in a sandbox, with one or both hands. This process is recorded with a fixed RGB-D camera, and afterwards the output data are collected into a data set which is intended for training neural networks. The study shows that the first assumption is valid, provided that mixed actions also exist, and that the second one has to be relaxed, because actions may affect, in some cases, to the entire state of the material. The assumption of a limited set of manipulation primitives is considered also by Ruggiero *et al.* [110], in the framework of the *RoDyMan* project. This project aims to develop new strategies in robotic nonprehensile dynamic manipulation of deformable objects, with a dual-arm anthropomorphic robotic platform for performing the challenging pizza-making process as the final demonstrator. The pizza-making process includes various deformation subtasks, which they divide into two different nonprehensile manipulation primitives: sliding and tossing. They show diverse results of the project that include a method for real-time tracking of the manipulated object, in which an RGB-D sensor is utilized for obtaining a point cloud that serves to create a finite element model of the object. From a different perspective, Simon and Basri [116] develop a shape matching method for finding a set of deformation actions such that the shape of an initially undeformed surface is transformed into a specific deformed configuration. The initial shape is discretized to a 2D linear elastic finite-element model, that is submitted afterwards to inner condensation in order to reduce the mesh nodes to the ones of the contour, while retaining the physical properties of the rest of the model. A non-linear optimization procedure allows to find the contour forces that produce the desired elastic deformation. It is important to remark that the method is locally defined, which makes it dependent on the source and target shapes initial alignment.

2.6.2 Transport actions

Transport actions are necessary when the manipulation task requires the deformable object to be placed on a position which differs from the initial one. High-level tasks usually imply to perform these kinds of actions at some specific instants of the process.

As mentioned before, decentralized control techniques are proposed for transporting a flexible payload by a multi-robot team in [16]. They limit deformation actions to the purpose of providing the required grasping force, while the transport actions have the main role in the tasks. Only the translational problem is considered, leaving the extension to the rotational case open. Also in Alonso-Mora's work the vast majority of the manipulation actions are transport ones. In [5] the considered manipulation tasks relegate deformation actions to avoiding obstacles and overstretching during the ensemble's motion. The same idea is developed by McConachie and Berenson in [94], due to the fact that the purpose is to move the object to a desired position while avoiding obstacles and maintaining the structural integrity of the manipulated piece of fabric.

2.7 Classification according to the application

Looking at the approaches from a more high-level point of view, a classification based on the practical application they are explicitly intended for can be carried out. While some methods develop general strategies that can be applied to many different purposes, others consist of a group of specific tasks with a well defined interest. We show here four of those specific applications to which some of the reviewed approaches can be associated.

2.7.1 Parts processing applications

Materials and parts processing is one of the most important applications in the industrial field due to its economical impact. For instance, meat processing is tackled in [88] with the aim of dividing large meat pieces into smaller ones. Manipulation of food products is also considered in [35] from the perspective of manipulation with shape preservation. In the field of electric and electronic components manipulation, one can find [142] and [14], where cable and flexible circuit boards manipulation are respectively studied.

2.7.2 Medical applications

The interest in techniques related to medical processes (surgery, inspections, etc.) is clear. These methods try to overcome some of the human limitations in terms of control, accuracy and accessibility to some parts of the body. Positioning and inserting elastic tissues in surgery are some of the possible applications of the methods in [102], [100] and [101]. Also in [2] heterogeneous phantoms are manipulated to perform surgery-related tasks, in dynamic environments where the phantoms' properties can change quickly.

2.7.3 Domestic applications

Assistive robotics is these days a source of robotic development that improves the perception that the society has towards robots. In the near future, the combination of assistive robotics with deformable objects manipulation will help people to perform some daily-life tasks, like cloth folding or tablecloth spreading. These tasks are succesfully simulated in [20], and in [86] cloth folding is complemented with garment type recognition. In [80] and [74] collaborative human-robot cloth manipulation is considered, with a system that decreases the amount of wrinkles produced by the human manipulator and a system that decreases the amount of wrinkles from a random initial state of the cloth respectively. Finally, the method in [84] could help people to perform cleaning tasks with delicate objects that require some amount of force for the stains to be removed.

2.7.4 Transport applications

There are some methods whose purpose is to move a deformable object from one place to another, and in those approaches the transport actions (see Section 2.6.2) are usually the most relevant ones. Mobile robots are well-suited for this kind of application, due to its ability to perform manipulation tasks along broad areas. This characteristic is exploited by Alonso-Mora *et al.* [5] to transport varied deformable objects in a wide space with obstacles. Bai and Wen [16], in turn, consider fixed robotic arms in experimental tests to manipulate the object in a very limited volume.

2.8 Classification according to the manipulator

In every manipulation strategy, a specific type of robotic manipulator must be selected to perform the involved tasks. This selection must be realized according to the method requirements and the operational limitations of the robots. However, some approaches simplify the robotic manipulators an treat them as grasped points, which move in a controlled way to accomplish the tasks.

2.8.1 Point manipulators

The most basic manipulator model is a point or a set of points that represent the position where the robot's end effector is attached to the deformable object. In many cases, the problem of controlling a deformable object is formulated without considering the manipulator's dynamics, and therefore the control action is directly a dynamic output to the grasped point. It may be useful to deal solely with the grasped points in the sense that substituting the robotic manipulator should not affect to the rest of the system. This fact is confirmed in [35], when the same approach is used to simulate a 2D shape control problem with multiple independent point actuators and a shape preservation problem with a multi-fingered robotic hand. Nevertheless, additional work must be done to apply this type of algorithm to a real environment, where the control outputs to the grasped points have to be transferred to the manipulator's own controller. Also in [130] a set of manipulation points is defined within the perimeter of the discretized deformable object. According to the authors, the motion of these points is controlled by a group of robotic fingers, but *a priori* the method would also be valid for the case of independent robotic grippers. In [34] and [14] the manipulation points correspond again to a predetermined set of nodes of the object model's mesh. Due to the fact that the method in [116] is designed to work within the context of computer vision and image recognition, the hypothetical "manipulators" that deform the shape are not represented. However, boundary forces are applied to the contour nodes of the finite element model of the object, which could be ultimately considered as point manipulators.

2.8.2 Fixed arm manipulators

Most of the reviewed methods consider standard robotic arms to perform the manipulation tasks. A rough description of this standard design might be a fixed to the ground segmented body, whose parts are linked by rotary joints and equipped with a gripping tool at its end effector. Independent fixed robotic arms work in collaboration in the following approaches: [102], [84], [16], [88] and [101].

When two fixed robotic arms are assembled into a common supporting structure, trying to reproduce the human body configuration, the ensemble is treated as a single dual-armed robot. These kinds of robots are utilized in [86], [74], [69], [142], [2], [80], [125] and [126].

2.8.3 Mobile arm manipulators

In general, a mobile arm manipulator consists in assembling a robotic arm on a wheeled platform, which drastically extends the action range of the robots. This is a quite recent concept of robot that contrasts with the widely adopted one of the confined fixed arm, and that is the reason why there are still not many industrial applications that include them. In [5] some robots of this kind are explicitly considered to carry out the manipulation process. Here, the action area of the agents is clearly much wider than the one of the previous section methods, and besides the robots can move over the floor in an omnidirectional manner thanks to the Mecanum wheels. Also the dual-arm robotic platform in [110] is mounted on an omnidirectional wheeled platform, so that it can perform the complex tasks involved in the pizza making process. A more simplified version of mobile robot is shown in [133], where a group of small low-cost manipulators move a box along the control points. Another interesting problem is tackled by Sreenath and Kumar [119], where a box is held in the air through cables attached to a group of quadrotors. The goal of the method is performing feasible trajectories with the system (payload and quadrotors), which is modelled by means of a hybrid dynamic model. They demonstrate that the hybrid system, which represents the behavior when all cable tensions are positive or one of them is zero, is differentially flat, and show in experimental tests the validity of the approach.

2.9 Multirobot control

For the purpose of developing new methods in manipulation of deformable objects by multiple robots, revising the general field of multirobot control can be undoubtedly helpful to find the suitable motion planning or task management strategies. In the present section, we analyze a brief selection of approaches that deal with multi-robot control problems, like robot formation control and task management, without considering explicitly the manipulation of deformable objects.

A distributed formation control technique is developed by Aranda *et al.* in [10]. The method allows a group of robotic agents to reach an specific target formation in 2D without the need of a common reference orientation or a formation leader. Based on the minimization of a Lyapunov function, the control law includes locally computed rotation matrices which are obtained by each robot by measuring the relative distances to the formation neighbors in its own local coordinate frame. Stability of the system is guaranteed by inducing an agents network that belongs to a certain class of undirected rigid graphs in 2D. These results are applicable to single-integrator and unicycle agents.

Alonso-Mora *et al.* propose a collaborative object transport strategy for dynamic environments in [4] in which a specific formation is maintained by the robotic team as the task proceeds. Two variants of the algorithm, which is scalable with the number of robots, are presented: a local formation planning method, in which large obstacle-free convex regions are obtained in the neighborhood of the robots, and a global path planning method, which extends the previous approach by connecting successive obstacle-free convex regions when a transition in formation is possible. The formation parameters (size, 3D-orientation, etc.) are obtained by solving a constrained optimization problem via sequential convex programming, and they are adjusted throughout the process in order to evade obstacles and avoid collisions. Individual controllers, which account for the dynamic behavior of each agent, allow the robots to navigate according to the planned trajectory. This method is validated in experimental tests, with a team of mobile ground manipulators, and also in simulations, with a team of micro aerial vehicles.

In [18] a different motion planning method is developed by Basile *et al.* from the perspective of task management in a hyper-flexible multi-robot workcell. User-specified motion at the workpiece level is provided to the global planner, which integrates a cooperative planner and an arm planner. While the former defines the cooperative task-space trajectory in terms of meaningful task variables, the latter translates the motion generated by the cooperative planner into joint space outputs. The reference workcell consists of a group of robotic arms that are divided in two different categories: positioners, which orient and displace the object, and workers, which perform the necessary operations in the manipulated object. A set of instructions that extends classical programming languages for industrial robots to multi-robot systems is derived from the developed formulation.

2.10 Discussion

From the analysis of the different methods that tackle the problem of manipulation of deformable objects by multiple robots, summarized in Table 2.1, some conclusions may be inferred. Overall, autonomous manipulation of deformable objects is an important and complex problem that is gaining attention in the recent years. It seems clear that using multiple robots is a necessary condition in order to perform the manipulation tasks with flexibility and robustness guarantees, and also to extend the workspace.

Focusing now on the studied solutions, although model-based approaches can have some advantages in terms of computational cost and accuracy, those approaches based on learning a deformation model in an online manner, or those that directly avoid considering a deformation model, are far more flexible and robust. This flexibility is due to the fact that less assumptions are made with respect to the nature of the object's material and the manipulation system. Concerning perception methods, hybrid force-vision systems should be chosen to obtain a more complete state of the deformable object. However, depending on the strategy or the system characteristics it may happen that only one type of feedback information is available.

With respect to the issues that remain unsolved in this context, they include: a complete and accurate perception of the deformable object, for any kind of shape; a complete integration between deformation control and transport of the object, with safety guarantees; and the ability to manipulate fragile objects without damaging them. These are the topics we tackle in this research work. Improving deformation sensing, the first of these aspects, is explored in the next chapter through a multicamera system for coverage of deformable shapes.

Table 2.1: Manipulation methods summary.

Reference	Model	Object	Control	Perception	Actions	Application	Manipulator
Das *et al.* [35]	Precomputed	2D (rheological)	Robust	-	Deform.	Parts processing	n point robots
Li *et al.* [86]	Precomputed	3D	-	Visual	Deform./trans.	Domestic	Dual-arm robot
Jia *et al.* [74]	Precomputed	2D	Classic	Visual	Deform./trans.	Domestic	Dual-arm robot
Duenser *et al.* [42]	Precomputed	3D	Optimal	-	Deform.	Domestic	Dual-arm robot
Long *et al.* [89]	Precomputed	1D-3D	-	-	-	-	n arms
Higashimori *et al.* [65]	Learned	3D (rheological)	-	Force-vision	Deform.	-	mobile plate
Navarro-Alarcon *et al.* [100]	Learned	2D,3D (elastic)	Adaptive	Vision	Deform.	Medical	cartesian robot and grippers
Navarro-Alarcon *et al.* [102]	Learned	1D-3D (elastic)	Adaptive	Vision	Deform./trans.	Medical	2 robotic arms
Hu *et al.* [69]	Learned	1D,2D	Nonlinear	Vision	Deform./trans.	-	Dual-arm robot
Berenson [20]	Learned	1D,2D	Jacobian	-	Deform./trans.	Domestic	2 floating grippers
Mcconachie *et al.* [94]	Learned	1D,2D	Jacobian	-	Trans.	Domestic	2 floating grippers
Langsfeld *et al.* [84]	Learned	2D (elastic)	-	Force-vision	Deform./trans.	Domestic	2 robotic arms
Zhu *et al.* [142]	Learned	1D	-	Vision	Deform.	Parts processing	Dual-arm robot
Alambeigi *et al.* [2]	Learned	3D	Optimal	Vision	Deform./trans.	Medical	n robotic arms
Wada *et al.* [130]	Model-free	2D	Classic	-	Deform.	-	n point robots
Bai *et al.* [16]	Model-free	3D	Adaptive	Force	Trans.	Transport	2 or more robotic arms
Long *et al.* [88]	Model-free	3D	Adaptive	Force-vision	Deform.	Parts processing	2 robotic arms
Kudoh *et al.* [82]	Model-free	1D	-	Vision	Deform.	Domestic	Dual-arm robot
Tang *et al.* [125]	Model-free	1D	-	Vision	Deform.	-	2 robotic arms
Tang *et al.* [126]	Model-free	1D	-	Vision	Deform.	-	2 robotic arms
Alonso-Mora *et al.* [5]	Model-free	1D-3D	Classic (low-level)	Force-vision	Trans.	Transport	n mobile arms
Kruse *et al.* [80]	Model-free	2D	Jacobian	Force-vision	Deform.	Domestic	Dual-arm robot and human
Navarro-Alarcon *et al.* [101]	Learned	3D (elastic)	Robust	Vision	Deform.	Medical	2 robotic arms
Sun *et al.* [122]	Precomputed	3D	Classic	Pose-velocity	Trans.	-	2 robotic arms
Hu *et al.* [68]	Precomputed	2D (elastic)	Robust	Deformation	Deform.	Smart structures	n piezoceramic actuators
Ruggiero *et al.* [110]	Precomputed	3D	Robust	Force-vision	Deform.	Pizza making	Dual-arm mobile robot
Dang *et al.* [34]	Precomputed	2D	Adaptive	Position & velocity	Deform.	-	n point robots
Delgado *et al.* [37]	Model-free	3D	Classic	Force	Deform./trans.	Domestic	Dual-arm robot
Simon *et al.* [116]	Precomputed	2D (elastic)	-	Force-vision	Deform.	-	n point robots
Sreenath *et al.* [119]	Precomputed	1D	Nonlinear	-	Trans.	Transport	n quadrotors
Mukadam *et al.* [99]	Model-free	2D (elastic)	Optimal	-	Deform.	-	n floating grippers
Asano *et al.* [14]	Precomputed	3D	-	-	Deform.	Parts processing	n point robots
Tokumoto *et al.* [129]	Precomputed	2D,3D	-	-	Deform.	-	n point robots

Chapter 3

Multicamera coverage of deformable shapes

This chapter describes our method for performing perception coverage of the contour of an object along a deformation process and according to a prescribed coverage objective, in terms of visibility and resolution. We propose new techniques for guaranteeing feasibility of the coverage objectives, which include the computation of the maximum visibility and resolution of the contour. Then, we solve an offline constrained optimization problem to obtain the minimum number of limited field-of-view cameras that achieve the coverage objectives. Alternatively, we describe an online technique that provides optimized configurations when the object's reference deformation is unknown. Our approaches are tested in simulated and real-world scenarios, in which our method achieves 100% of the target coverage.

3.1 Introduction

Measuring deformation of objects whose shape changes over time is, as outlined by the previous analysis of the state of the art, an effective but challenging means for evaluating the quality and the level of execution of tasks where deformable objects are autonomously manipulated. This is particularly true when dealing with objects that are large, fragile or that show complex shapes or small deformations [126]. In these cases, control approaches require a continuous and complete feedback of the object's state [35], [110], which allows tracking the object's shape and evaluating the overall performance of the manipulation process. Such control and feedback systems usually employ multiple agents and sensing units [58], [27], [127]. Besides the industrial interest of these perception systems, medical applications related to robot-assisted surgery also demand accurate and flexible sensing solutions [100], [3].

We focus our interest in industrial tasks where the shape of an object is deformed following a prescribed guideline. This is the case for manufacturing processes that transform raw materials (cloths, plastics...) into objects with specific shapes. Guaranteeing a continuous feedback of the object's state in industrial environments may be challenging, taking into account the highly dynamic and complex structure of deformable objects and the external perturbations. This is why our research problem consists in developing a multi-camera perception system to cover an object whose shape changes over time, according to a set of prescribed requirements. Coverage is defined in terms of completion and accuracy of the detected object's contour. We also want to optimize the performance of the perception system by minimizing the number of sensing resources. The main properties we exploit to define the coverage task are visibility, as a measure of coverage completion, and resolution, as an accuracy indicator. A potential application of our system is the inspection of 3D printed parts, which are created by stacking layers of material. Our techniques could be applied to cover the lateral surface of the parts as they are built, to discover potential cracks and check the quality of the interlayer bonding. The usefulness of this proposal is also significant with tasks involving objects of great dimensions, such as the manipulation of large plastic containers, in which the lateral surface of the contour must be continuously supervised for quality purposes. Other example application would be the inspection of high precision machined parts during manufacturing. These kinds of parts have critical cross sections whose lateral surfaces must be controlled carefully to prevent defects. Our proposal could also be applied to monitor assembly tasks [61].

The perception of areas of interest, according to different requirements and constraints, includes techniques that range from coverage to tracking. Depending on the type of deployment of the sensors, we can classify these approaches as static or dynamic. The deployment criterium is closely related to whether the perceived environment is time-dependent or not. Particularly, statically positioning the sensors is usually an appropriate strategy when there are no substantial changes over time in the perceived environment. The well-known art-gallery problem is essentially a static coverage strategy where the number of surveillance cameras is minimized. Recent developments include variants of this problem for tracking of mobile intruders with fixed and mobile guards [45]. In general, those studies based on the general art-gallery problem do not include realistic models with sensing constraints [109]. This problem also shares some aspects with the one we tackle, but classical solutions to the art-gallery problem are not directly applicable to our time-varying scenario. Other static perception strategies with multiple sensors are selected for purposes like industrial inspection of 3D parts [138], 3D reconstruction of pipelines for inspection [137], buildings exterior surveillance [120], volume estimation of merchandise [13] and pose estimation in restricted space scenarios [39]. These kinds of works select the viewpoints with *ad hoc* criteria or from a predefined candidate set, which may be non-optimal, and have limited adaptability. In contrast, our approach computes at every deformation instant the optimal set of cameras in the continuous space, parameterized in polar coordinates (angle and radius).

Approaches where there is a single static sensing unit have been proposed for estimating the deformation properties of deformable objects [52]. In other cases, dynamic deployment may be necessary to obtain complete information of the area of interest. This happens in coverage systems of 3D rigid parts for industrial inspection tasks [77] and in structure from motion approaches [38]. Our target problem shares common points with the structure from motion topic, but in that case the environment is static in general [54], not dynamic as ours. Dynamic deployment of multiple sensing units may also be required for full perception at every time instant. This is the case in centralized approaches where small teams of robots monitor 3D static environments [36] or enclose and track a mobile target [90], and also in decentralized strategies where 2D static environments are monitored by large teams of robots [113].

As for the active perception field [107], [17], works in this domain are dynamic in general. Multi-camera centralized networks, where the position and orientation of each sensor are optimized, are considered in studies of this field for shape recovery of moving deformable objects [112], [104]. Compared to our approach, these works do not minimize the number of cameras that are necessary to recover the shape of the object.

In an earlier version of this work [59], we have described a new technique for obtaining the maximum visibility of a given shape from a certain distance to it. Besides, we proposed a new formulation over a continuous search space, with safe and robust perception criteria, and the minimization of the number of cameras required for the coverage task. The present chapter improves and extends the previous work with the following main contributions: (a) a new formulation of the optimization problems, which is less constrained and more efficient; (b) the computation of the maximum angular resolution of a shape; (c) a fast coverage technique based on local optimization of the cameras; and (d) a novel analysis of the geometrical visibility properties of a shape. The improved system is evaluated with additional results and analyses from different experiments, in both simulated and real scenarios.

3.2 Problem statement

3.2.1 Framework and assumptions

Let us consider an object that undergoes non-negligible deformations due to some specific manipulation process. The object's shape is defined by the 2D contour of the object's vertical projection onto the ground plane. For inspection purposes, the contour must be sampled at each instant k ($k = 1, ..., K$) by a set of 3D cameras. These cameras $\mathbf{c}_i(k)$ ($i = 1, ..., C$) are able to position and orient themselves around the contour's centroid $\mathbf{g}(k)$ (e.g. they are mounted on mobile robots or robot arms). In particular, their degrees of freedom are the following: (i) $\psi_i(k)$, the angle around $\mathbf{g}(k)$ where the center of the camera is located, (ii) $d_i(k)$ ($d_{min} \leq d_i(k) \leq d_{max}, \ \forall i$), the distance between the camera center and $\mathbf{g}(k)$, and (iii) $\phi_i(k)$, the camera orientation. We consider that the cameras' movements obey single-integrator kinematics, and no vision model is selected *a priori*. Instead, we focus on ray tracing from the optical center of the cameras, and angular and range parametric constraints are applied to the rays to approximate the restricted Field of View (FOV). This formulation is compatible with any model based on ray tracing from an optical center, like laser-based sensors, or even with tools like spray guns. Figure 3.1 shows the overview of the studied system.

The different assumptions we consider in the present study are reported next:

Assumption 3.1. ***Known data.*** We consider the following:

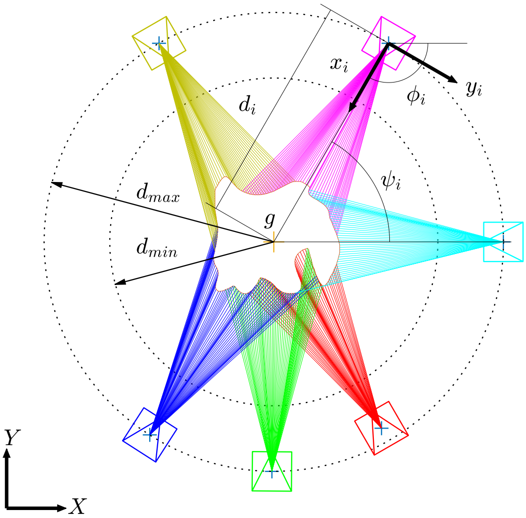


Figure 3.1: Overview of the system we consider. At the center of the figure, the 2D sampled contour of the object is fully covered by six restricted FOV cameras. As the object deforms, the cameras will synchronously move and rotate so that the coverage is guaranteed at all times. The global axes (X, Y) and the local optical axes of camera i (x_i, y_i) are represented. In addition, we show the projection rays that link the center of each camera to the contour vertexes.

1. Cameras can get accurate 2D point clouds of the environment.
2. We know the state of the cameras with respect to the object $\mathbf{c}_i(k) = [\psi_i, d_i, \phi_i]$ at any time.
3. We know the desired deformation of the object. Therefore, we have at each instant k an approximated shape of the 2D contour of the object, called reference contour, as well as an approximated $\mathbf{g}(k)$.

Assumption 3.2. ***Slow deformations.*** We assume that the object's deformations (i.e. contour shape variations) between each two consecutive deformation instants k and $(k+1)$, $\forall k$, are small. We assume, therefore, that the dynamics of the deformable object are as slow as to allow the cameras to reach their required positions and orientations $\forall k$.

It is not unusual to consider reference models that complete missing information for reconstructing complex setups [137] or deformable objects [9]. In our case, the reference contour gives the desired shape of the object during the deformation process. For example, for inspection of 3D printed parts, the desired contour shape of the 2D layers, which is known *a priori*, would represent the reference contour. Depending on the task, it does not need to be accurate necessarily. If the purpose is, for instance, to inspect a large object, a rough approximation of the contour may suffice. A collection of reference segments is obtained by sampling the reference contour, and the number and size of the reference segments $\mathbf{s}_j(k) = \left[\mathbf{v}_j, \mathbf{v}_{(j+1)}\right]$ $(j = 1, ..., S)$ must be set according to the requirements of the inspection task and the complexity of the contour shape.

For clarity purposes, we will omit in general the (k) term in the rest of the chapter.

3.2.2 Optimization objectives

We set the coverage objectives, in terms of visibility and resolution, from an optimization point of view.

Definition 3.1. Visibility cost. *We define the visibility cost*

$$\gamma_v = \frac{\sum_{j=1}^{S} \gamma_{v_j}}{\sum_{j=1}^{S} \vartheta_j^*} \text{ , with } \gamma_{v_j} = \left\{ \begin{array}{cc} \vartheta_j^* - \vartheta_j & \text{if} \quad \vartheta_j^* > \vartheta_j \\ 0 & \text{otherwise} \end{array} \right. . \tag{3.1}$$

ϑ_j indicates the number of cameras that detect segment $\mathbf{s}_j$ at the k instant, and ϑ_j^* represents the target number of cameras that must detect $\mathbf{s}_j$ according to the requirements of the inspection task.

The *visibility cost* describes how far the system is from the objective of detecting the contour segments as required, in terms of visibility. Note that detecting a segment with more cameras than necessary does not contribute to decrease the value of the *visibility cost.*

Definition 3.2. Resolution cost. *We define the resolution cost*

$$\gamma_r = \frac{1}{\beta \sum_{j=1}^{S'} 1/r_j^*} \sum_{j=1}^{S'} \frac{\beta - r_j}{r_j^*} \quad , \tag{3.2}$$

where β is the cameras' angle of view, r_j indicates the maximum angular resolution of segment $\mathbf{s}_j$, r_j^* is the target angular resolution of $\mathbf{s}_j$ and S' is the number of visible segments with $r_j^* > 0$.

The *resolution cost* allows to evaluate the angular resolution of the segments with respect to the maximum resolution. The target resolution should take higher values in those areas where more accurate perception is required, as it will drive the cameras towards front-facing positions with respect to the segments. In case no specific resolution is required in an area, the segments there should be assigned $r_j^* = 0$.

As a consequence, the near-optimal configurations of the cameras $\mathbf{c}_i^* = [\psi_i^*, d_i^*, \phi_i^*]$ are the ones that reduce both costs to, at least, these values:

$$\begin{aligned} \gamma_v^* &= 0 \, , & (3.3) \\ \gamma_r^*(r_j = r_j^* \text{ if } r_j > r_j^*) &= \beta \sum_{j=1}^{S'} \left(\frac{1}{r_j^*} \right) - S' \, . & (3.4) \end{aligned}$$

Then, the problem we tackle consists in configuring a minimal set of cameras in a near-optimal manner so that they cover a 2D contour that deforms over time, as specified in terms of visibility and angular resolution. Inter-camera collisions and occlusions must be avoided to guarantee coverage in the whole deformation process. Note that placing a single camera on top of the contour for inspection is not a valid solution in our case. This is due to the fact that lateral views of the contour would not be obtained for inspection.

Properly defining the target visibility and angular resolution is crucial for obtaining the desired behavior of the system. In the next section, some tools are proposed for defining the target visibility and angular resolution with feasibility guarantees.

3.3 Maximum visibility and resolution

In order to define feasible objectives we need to identify the segments that are inevitably occluded, and also the maximum angular resolution of each detected segment that a camera placed around $\mathbf{g}$ can obtain.

Definition 3.3. Maximum visibility. *The maximum visibility* ϑ_j^{max} *of a segment is a value that indicates the maximum number of cameras that detect* $\mathbf{s}_j$ *from the distance d. If* $\mathbf{s}_j$ *can be detected,* $\vartheta_j^{max} > 0$*, and* $\vartheta_j^{max} = 0$ *otherwise.*

The *maximum visibility* of convex contour shapes is known *a priori* in an obstacle-free framework ($\vartheta_j^{max} > 0, \ \forall j$). However, when dealing with non-convex shapes non-visible zones may appear due to auto-occlusions.

Previously to computing this property, we need a way to determine whether a segment $\mathbf{s}_j$ can be detected or not from certain camera position. For this purpose, we consider a system based on *bi-partite visibility graphs*.

Definition 3.4. Bi-partite visibility graph. *The bi-partite visibility graph*

$$G_v = (\mathbf{C}_T, \mathbf{S_{ref}}, \mathbf{E}) \ , \tag{3.5}$$

$$\mathbf{C}_T = \{\mathbf{c}_1, ..., \mathbf{c}_i, ..., \mathbf{c}_C\}, \ \mathbf{S_{ref}} = \{\mathbf{s}_1, ..., \mathbf{s}_j, ..., \mathbf{s}_S\} \ , \tag{3.6}$$

is the graph where an edge $\mathbf{e}_{i,j} \in \mathbf{E}$ *connects the camera center* $\mathbf{c}_i$ *with the vertex* $\mathbf{v}_j \in \mathbf{s}_j$ *if the virtual line that links them in the 2D space does not intersect any obstacle, i.e. if* $\mathbf{v}_j$ *is visible from* $\mathbf{c}_i$*. Each edge's weight is equal to the Euclidean distance* $d_E(\mathbf{c}_i, \mathbf{v}_j)$*.*

Then, if two edges $\mathbf{e}_{i,j}$ and $\mathbf{e}_{i,j+1}$ exist, such that $\mathbf{e}_{i,j} = [\mathbf{c}_i, \mathbf{v}_j]$ and $\mathbf{e}_{i,j+1} = [\mathbf{c}_i, \mathbf{v}_{(j+1)}]$, we determine that segment $\mathbf{s}_j = [\mathbf{v}_j, \mathbf{v}_{(j+1)}]$ is visible from $\mathbf{c}_i$. Figure 3.2 shows a visible segment $\mathbf{s}_j$ with its angular resolution r_j. We formulate next the method to compute the *maximum visibility* of a 2D discretized contour from a prescribed distance to $\mathbf{g}$.

Proposition 3.1. Consider the *bi-partite visibility graph* of an scenario with no obstacles where the object is surrounded by infinite omnidirectional cameras ($C \to \infty$) with unlimited FOV

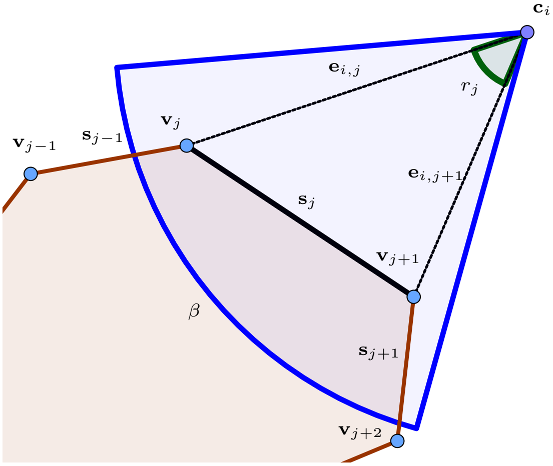


Figure 3.2: Example of visible segment. Segment $\mathbf{s}_j$ is visible from $\mathbf{c}_i$ with angular resolution r_j (in green), under the angle of view β (in blue).

range. The cameras completely cover a circumference of radius d centered at $\mathbf{g}$. In such situation, the graph provides exactly the number of segments that are visible from that circumference, with $\vartheta_j^{max} \to \infty$ if $\mathbf{s}_j$ is visible and $\vartheta_j^{max} = 0$ otherwise.

Proof. By definition 3.4, $\mathbf{s}_j$ is visible from $\mathbf{c}_i$ if its two vertexes, $\mathbf{v}_j$ and $\mathbf{v}_{(j+1)}$, are connected by edges to $\mathbf{c}_i$. Thus, ϑ_j^{max} of the setup described in Proposition 3.1 can be computed as follows:

$$\vartheta_j^{max} = \sum_{i=1}^{C} (\exists \mathbf{e}_{i,j} \wedge \exists \mathbf{e}_{i,j+1}) \ , \tag{3.7}$$

where we treat the boolean outputs *true* and *false* as integers 1 and 0, respectively. Given that every possible sensor location at a distance d around $\mathbf{g}$ is considered and that omnidirectional cameras are able to detect in any possible direction, every possible valid edge will be determined. Therefore, every visible segment will be obtained with $\vartheta_j^{max} \to \infty$, as well as the occluded segments with $\vartheta_j^{max} = 0$. □

Remark 3.1. *Note that in practice the number of cameras C is a finite value. For this reason, an exact accuracy for identifying occluded segments, when computing ϑ_j^{max}, may not be achieved (infinite positions for detection are not evaluated). Despite this, we obtain a useful measure: the closer ϑ_j^{max} is to C, the wider the range of possible locations from which $\mathbf{s}_j$ can be detected. This measure allow us to know the difficulty of detecting a segment compared to the rest.*

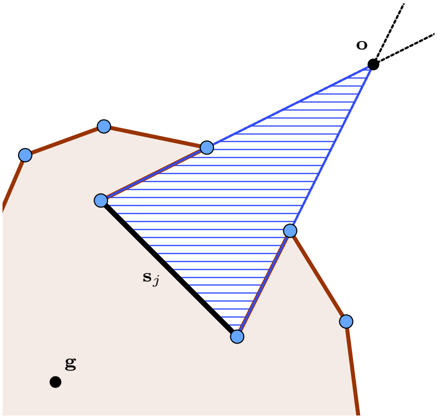


Figure 3.3: Example of contour shape with a dovetail slot. Segment $\mathbf{s}_j$ will be fully detected only by those cameras that are placed within the blue dashed area.

Next Proposition includes additional reasoning about the *maximum visibility*, concerning the distance from which this one is obtained.

Proposition 3.2. Consider a *bi-partite visibility graph* with infinite omnidirectional and unlimited FOV range cameras, which are placed at a distance d from $\mathbf{g}$. Then, by reducing the distance d the visibility of the contour segments will remain constant or it will increase for any possible shape, but it will never decrease.

Proof. If we consider the paradigmatic case of a contour that includes a dovetail shaped slot, whose inner part consists of a single segment $\mathbf{s}_j$, only those cameras that are placed inside an specific area in front of the slot will be able to detect the segment. This area is contained between $\mathbf{s}_j$ and the two adjacent lines that start at both vertexes of $\mathbf{s}_j$ and intersect at the point $\mathbf{o}$ (Fig. 3.3). Thus, only if $d \leq \overline{\mathbf{g}\,\mathbf{o}}$ the segment $\mathbf{s}_j$ will be detectable. In addition, no geometric construction exists such that the opposite effect occurs: the area in which the camera must be placed to detect the segment will always start at the segment, and it will extend from there on. □

Remark 3.2. *An interesting observation arises when applying the test of Proposition 3.1 to a continuous contour, e.g. given by a parametric curve* $\{x(\psi), y(\psi)\}$. *As long as the infinite cameras lie outside the convex hull of the contour,* ϑ_j^{max} *will remain constant for every segment independently of the distance d. This is explained by the fact that when obtaining* ϑ_j^{max}, *each edge connecting a camera center to a contour vertex will determine a semi-infinite projection ray. These rays start at the contour vertexes and have the direction of the edges, and their main property is that any camera placed on them will be able to detect the contour vertex. Given that the cameras can be placed on any circumference outside the convex hull centered at* $\mathbf{g}$, *and that all projection rays cross the convex hull, any circumference of infinite cameras will intersect all projection rays. Thus,* ϑ_j^{max} *values will remain constant independently of d.*

Definition 3.5. Maximum resolution. *The maximum resolution r_j^{max} of a segment is a value that indicates the maximum angular resolution of $\mathbf{s}_j$ that can be obtained from the distance d to $\mathbf{g}$ by a camera. If $\mathbf{s}_j$ can be detected, $r_j^{max} > 0$, and $r_j^{max} = 0$ otherwise.*

The setup described in Proposition 3.1, which allows us to obtain the *maximum visibility*, is also appropriate without modifications for obtaining the *maximum resolution* in the following manner:

$$r_j^{max} = \max_i \left[\cos^{-1} \left(\frac{\mathbf{e}_{i,j} \cdot \mathbf{e}_{i,j+1}}{\|\mathbf{e}_{i,j}\|_2 \cdot \|\mathbf{e}_{i,j+1}\|_2} \right) \right] . \tag{3.8}$$

Note that the shorter the distance d, the greater r_j^{max} values are.

The *maximum visibility* and the *maximum resolution* establish upper bounds that cannot be surpassed in terms of visibility and resolution of the object's contour. It is important to mention that we consider point cameras when computing the *maximum visibility* and *the maximum resolution*. This may imply that the real system, with physical constraints, cannot obtain the maximum values for all the segments at the same time.

3.4 Offline optimization

3.4.1 Limits of the number of cameras

In addition to the maximum visibility and resolution, we are going to compute the lower and upper bounds of the number of cameras to perform the required coverage. The minimum number of cameras can be obtained with additional formulation from the setup described in Proposition 3.1. Approaches dealing with coverage problems often include a visibility map or visibility function, which relates the point of view of a sensor with the parts of the object it is covering at certain time instant [120], [104]. From Proposition 3.1 we can also consider a visibility matrix $\mathbf{V}_{S \times C} = [V_{ji}]$ that indicates which segments are detected by each camera, i.e. $V_{ji} = 1$ means that segment $\mathbf{s}_j$ is visible for the camera $\mathbf{c}_i$, and $V_{ji} = 0$ otherwise. Then, we define the following integer linear programming problem:

$$\begin{aligned} \text{Given} \quad & \mathbf{V}, [\vartheta_1^*, \vartheta_2^*, ..., \vartheta_S^*]^T \\ \text{minimize} \quad & \sum_{i=1}^{C} \epsilon_i \\ \text{subject to} \quad & \mathbf{V} \cdot [\epsilon_1, \epsilon_2, ..., \epsilon_C]^T \geq [\vartheta_1^*, \vartheta_2^*, ..., \vartheta_S^*]^T \ , \\ & \epsilon_i \in \{0, 1\}, \ \forall i \ , \end{aligned} \tag{3.9}$$

where $\epsilon_i = 1$ *activates* the camera $\mathbf{c}_i$ and $\epsilon_i = 0$ *deactivates* it. The final cost, after solving (3.9), represents the minimum number of omnidirectional cameras, with unlimited range, placed at a distance d from $\mathbf{g}$ that achieve the target visibility. Here we only consider the visibility objective, so that we get a more conservative solution than considering both visibility and resolution.

As for obtaining the upper bound of the number of cameras, we follow a conservative area division strategy. We compute the maximum number of cameras as

$$\text{truncation}\left(\frac{d_{max}^2 - d_{min}^2}{d_c^2/4}\right) \geq C \ , \tag{3.10}$$

where radii d_{min} and d_{max} represent the space where the cameras are allowed to move, and $d_c/2$ is the radius of the minimum bounding circle of a camera. Essentially, we obtain with this equation the maximum number of cameras fitting in the allowable space in an approximate but conservative manner.

3.4.2 Minimization problem

Once we know how to set feasible ϑ_j^* and r_j^*, and also the range where to search the minimum number of cameras, we can define the optimization problem at instant k we are interested in.

Firstly, we define a new type of *bi-partite visibility graph* with the purpose of including the FOV constraints of the cameras into the problem.

Definition 3.6. Restricted visibility graph. *The restricted visibility graph is a bi-partite visibility graph where all edges connecting a camera with the contour vertexes are contained within the cameras FOV limits β and L_{max}.*

This new graph is created by comparing the weight of each edge $\mathbf{e}_{i,j}$, given by the Euclidean distance $d_E(\mathbf{c}_i, \mathbf{v}_j)$, with the cameras' maximum range L_{max}, and the angle between the edge and the camera's local axis x_i (Fig. 3.1) with the cameras' angle of view β. If the weight is greater than L_{max} or the angle is greater than $\beta/2$, we consider that $\mathbf{v}_j$ is not visible from $\mathbf{c}_i$, and therefore we remove the edge from the graph. After the comparison and filtering, the intersections checking is applied to the remaining edges. In this case, the set of obstacle segments includes not only the set of segments of the object $\mathbf{S_{ref}}$, but also the set of camera segments. This is due to the fact that the FOV of a camera may be occluded by other neighboring cameras. The set of camera segments is obtained by sampling the contour of the geometric 2D camera model, which represents the 2D shape of the cameras.

With these new constraints for computing ϑ_j and r_j, the optimization problem is defined as follows:

$$\begin{array}{rl} \text{Given} & \mathbf{C}_T, \mathbf{S_{ref}} \\ \underset{\mathbf{C}_T}{\text{minimize}} & \gamma = F \cdot \gamma_v + (1-F) \cdot \gamma_r + \gamma_\sigma \\ \text{subject to} & d_E(\mathbf{c}_i, \mathbf{c}_l) \geq d_c \ , \\ & \forall i, l = 1, ..., C, \ i \neq l \ . \end{array} \tag{3.11}$$

γ is the cost function of the problem, that includes a linear combination of γ_v and γ_r through the interest factor $F \in [0,1]$. F is a user-defined constant that regulates the relative contribution of the visibility and resolution terms, as required by the specific coverage task. The remaining term, γ_σ, exploits the *potential field approach* to prevent collisions in those scenarios that require two cameras to be close to each other:

$$\gamma_\sigma = \frac{1}{(2\,\min_{\forall i,l,\, i \neq l}\{d_E(\mathbf{c}_i, \mathbf{c}_l)\}/d_c - 1)^w} \tag{3.12}$$

where the 'min' operator selects the minimal distance between two neighboring cameras, d_c is the minimal safety distance between two neighboring cameras, and w is an exponent that regulates the distance where γ_σ is effective. This exponent $w > 0$ is tuned according to the safety requirements, so that w is small when large separations are demanded, and large otherwise. Thus, the γ_σ term introduces flexibility for collision avoidance, with a positive contribution that shows near zero values when the cameras are widely spread and near 1 values when the distance between two of them approaches to d_c. In addition to γ_σ, the set of non-linear constraints that we include at the end of (3.11) prevents two cameras from getting closer than d_c under any circumstances. These constraints also avoid considering near-infinite values of γ_σ that can lead to a malfunctioning of the *pattern search* optimizer.

We propose an iterative technique to solve (3.11) at the deformation instant k. In the first iteration the problem is solved for the minimum number of cameras computed in (3.9), and the values of γ_v and γ_r are checked. If values in (3.3) and (3.4) are satisfied, the process stops, and the near-optimal configurations $\mathbf{C}_T^*$ and the near-optimal number of cameras C^* are obtained. Otherwise, the problem is iteratively solved for $C := C + 1$ until equations (3.3) and (3.4) are satisfied or the maximum number of cameras from eq. (3.10) is reached. We solve (3.11) with the *pattern search method* [15], which is a derivative-free approach compatible with our cost function γ. Although global optimality is not guaranteed, this method always provides, at least, a local optimum.

3.4.3 Minimization problem of the complete deformation process

We extend the result obtained in the previous section, the minimum number of cameras $C^*(k)$ at the deformation instant k, to the complete deformation process. The first step consists in solving (3.11) for a set of instants such that Assumption 3.2 is still satisfied at each two consecutive instants. When we have the solution for each selected k, we compute the minimum number of cameras that are required to perform the coverage task as $C^* = \max_{\forall k} C^*(k)$. This conservative measure improves the robustness against unexpected situations, although it may overfeed the system with unnecessary cameras at certain instants. After that, the optimization is performed again with $C(k) = C^*, \forall k$, and the reference positions and orientations are computed. The main steps involved in the minimization of the number of cameras for the full deformation process, and the sequence in which they are performed, are shown in Algorithm 3.1.

It is important to highlight that this process is offline, i.e. it is performed over the reference contours and before the real task takes place. If the evolution of the object's deformation is similar to the planned one during the real coverage task and the optimization objectives are still achieved, these reference configurations are the ones the cameras will adopt. The target

Algorithm 3.1 Compute C^*.

```
Set $k \leftarrow 0$
$C^* \leftarrow C_{min}(k)$ # obtained from (3.9)
while $k \leq K$ do
    Initialize $\gamma_v > \gamma_{v_{stop}}$ and $\gamma_r > \gamma_{r_{stop}}$
    $C^*(k) \leftarrow C_{min}(k) - 1$
    while (($\gamma_v > \gamma_v^*$) or ($\gamma_r > \gamma_r^*$)) and ($C(k)^* < C_{max}$) do
        # $C_{max}$ is obtained from (3.10)
        $C^*(k) \leftarrow C^*(k) + 1$
        Minimize $\gamma(k, \psi_i, d_i, \phi_i)$ # defined in (3.11)
    end while
    if $C^* < C^*(k)$ then
        $C^* \leftarrow C^*(k)$
    end if
    $k \leftarrow k + 1$
end while
return $C^*$
```

visibility and resolution are evaluated online by comparing the detected segments with the ones in $\mathbf{S_{ref}}$. During the real coverage task, if one of the points detected by the cameras, at least, lies on an area containing a reference segment, we consider that the reference segment is detected. The accuracy of the selected cameras determines the dimensions of this area. Then, if equations (3.3) and (3.4) are satisfied at k, each camera will follow the shortest path from $\mathbf{c}_i(k)$ to $\mathbf{c}_i(k+1)$, while avoiding collisions with the neighboring cameras to reach the next optimal configuration. In case deformation evolves in a different manner than the planned one, we propose in the next section an online reactive strategy that improves the achievement of the optimization objectives in a setup with high uncertainty.

3.5 Online optimization

During the real coverage task the object may undergo unexpected deformations. In order to properly capture these deformations and continue to cover the object as required, the system must be able to reconfigure itself in an online manner if necessary.

In case abnormal deformations (over a threshold) are detected, the object's reference contour is updated with the cameras data. Then, we propose an online repositioning technique that performs local optimizations at the camera level, as opposed to the offline method, where optimizations occur at the set level. This method provides near-optimal configurations of lower quality than the global optimization, but much faster than the former one. According to Assumption 3.2, deformations are small between k and $(k+1)$, and therefore we consider that $\mathbf{c}_i(k)$ and $\mathbf{c}_i(k+1)$ will be close. Then, we can restrict the values of the optimization pa-

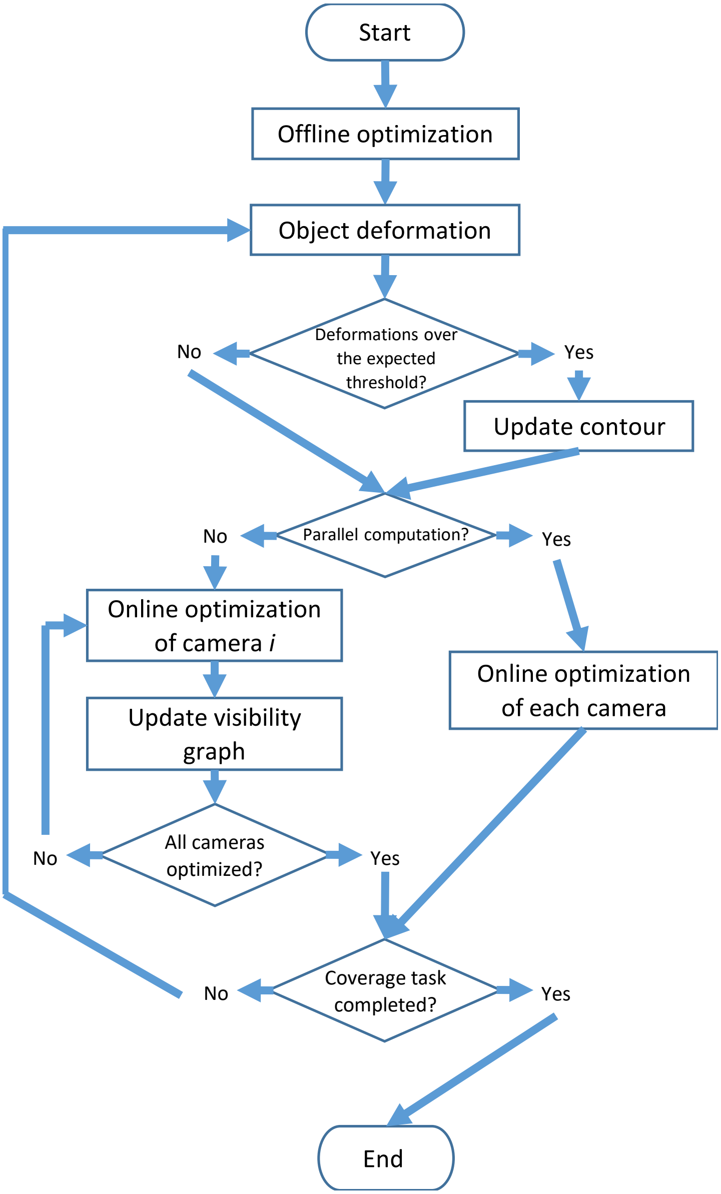


Figure 3.4: Flow chart including the main elements in the online optimization method.

rameters and perform individual optimizations with each camera. The new local optimization problem at the deformation instant $(k+1)$ is defined as follows:

$$\begin{aligned} \text{Given} \quad & \mathbf{C}_T,\ \mathbf{S} \\ \underset{\mathbf{c}_i}{\text{minimize}} \quad & \gamma = F \cdot \gamma_v + (1-F) \cdot \gamma_r + \gamma_\sigma \\ \text{subject to} \quad & d_E(\mathbf{c}_i(k+1), \mathbf{c}_l(k+1)) \geq d_c\ , \\ & \mathbf{c}_i(k) \cdot (1-m) \leq \mathbf{c}_i(k+1) \leq \mathbf{c}_i(k) \cdot (1+m)\ , \\ & \forall i, l = 1, ..., C,\ i \neq l\ , \end{aligned} \tag{3.13}$$

where $\mathbf{S}$ is the set of detected segments and m is a user-defined threshold that sets the allowed margin to perform the local optimization. It is worth mentioning that, despite the fact that only the configuration of one camera is optimized each time, the complete visibility graph with all the cameras is considered. The local optimizations can be solved in parallel, but better results in terms of optimality can be obtained if the problem is solved sequentially for each camera, at the expense of a higher computational cost. After each local optimization, the visibility graph is updated with the newly computed configurations.

The local optimization method also allows to relax Assumption 3.1.3 as follows: The deformation of the object is unknown, but an approximated shape of the 2D contour of the object, called reference contour, as well as an approximated $\mathbf{g}(k)$ are known *at the initial deformation instant* $k = 1$.

This means that we can utilize this technique even when the deformation evolution is unknown *a priori*, provided that the initial contour shape is known. As long as the detection of segments with coverage requirements ($\vartheta_j^* > 0$ or $r_j^* > 0$) is maintained from k to $(k+1)$, the local optimization can be directly applied over the updated contour. At the initial deformation instant we still need to apply the iterative optimization, in order to obtain the set of C^* minimal cameras with their initial near-optimal configurations. C^* will remain constant for the whole deformation process, as well as the target visibility and resolution. Note that maintaining the target values over time requires tracking and matching of the contour. We consider that the development of these techniques is out of the scope of this study, and we assume they are given. Figure 3.4 shows a flow chart with the main processes of the online optimization method.

It may happen that the local optimizations are not able to achieve the target visibility and resolution at some k. In such situations, the global optimization, with the complete set of cameras, can be executed online.

3.6 Experimental results

In this section, we present different experiments focused on validating the theoretical results and techniques we have developed. We evaluate in Matlab® a general set of cases, with diverse shapes and deformation processes. In the absence of a specific data set with 2D objects that undergo deformation, the 2D shapes we use in the simulations have been obtained from the MPEG-7 data set [85]. This data set contains many different images in bitmap format. From these images, we have extracted a sampled contour by using a regularly spaced grid. Firstly, we will test the effectiveness of our methods for obtaining the desired results and behaviors, i.e. how well they meet the expected operational requirements. Secondly, we will evaluate their performance and utility for being applied on real perception tasks.

3.6.1 Maximum visibility and resolution tests

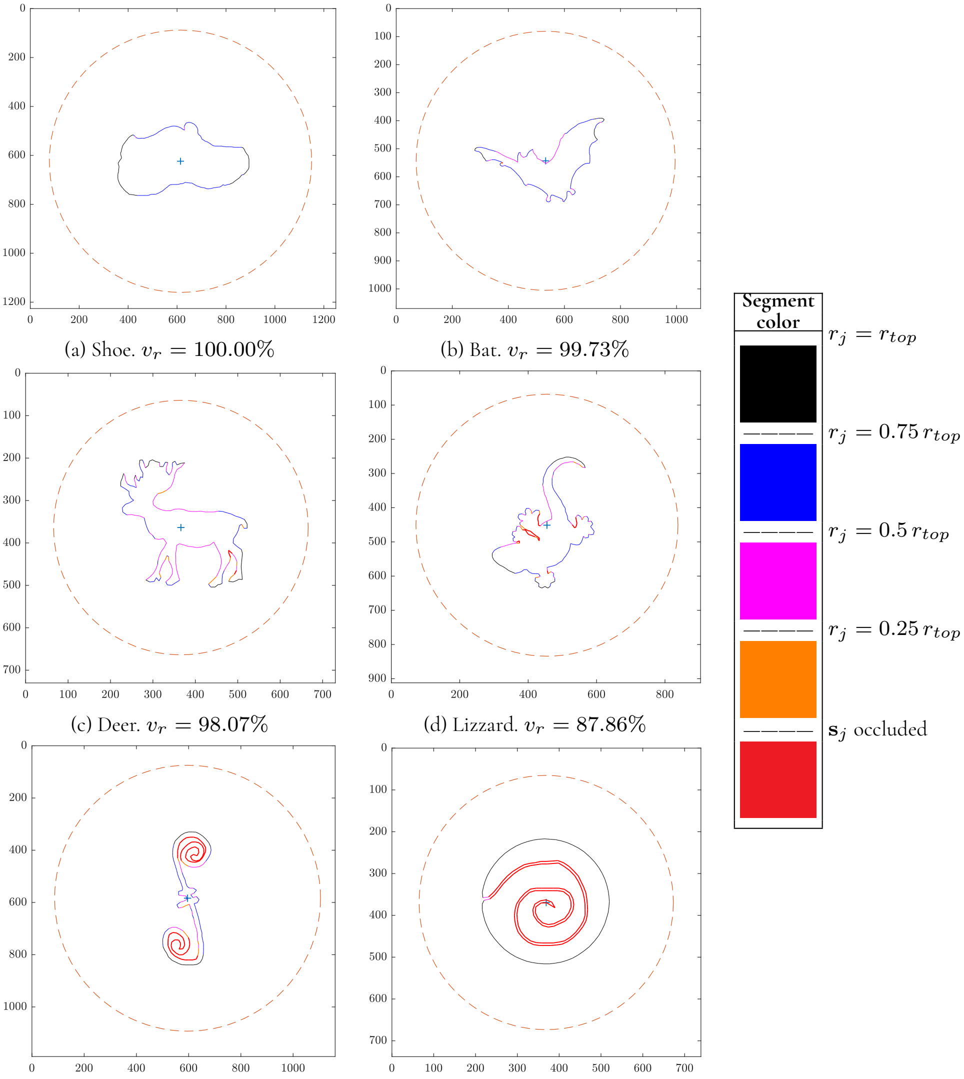
We will start by testing the effectiveness of our methods for obtaining the *maximum visibility* and *maximum resolution* of contours from the data set. As we previously mentioned, this information is needed for setting feasible coverage objectives, and it will be exploited in the next sections. Given a 2D image from the data set, we extract and sample the contour to create a set of reference segments $\mathbf{S_{ref}}$. Then, we cover the sampled contour with a set of evenly spaced omnidirectional cameras with unlimited FOV range at a distance d from $\mathbf{g}$, and afterwards we create and analyze *bi-partite visibility graph*. We have applied the method to the 1400 images of the MPEG-7 data set with the following common parameters, for uniformity purposes: $S = 200$ segments, $C = 500$ cameras and d is equal to the greater side of each contour bounding box. Figure 3.5 shows some examples that have been tested with our method, including their visibility ratios v_r, defined as

$$v_r(\%) = 100 \;\cdot\; visible\,perimeter \;/\; total\,perimeter \;. \tag{3.14}$$

The segments are colored depending on their maximum resolution r_j with respect to r_{top}, the absolute maximum resolution, in five different intervals. Figure 3.6 shows a box plot of the v_r values of all the images from the data set, grouped in six different categories according to the total contour perimeter. Results seem reasonable for different reasons. As expected, lower visibility ratios are obtained as the contour perimeter increases, since shapes with longer perimeters in the data set are usually more complex. We can see also that as the contour perimeter increases, so does scattering in the v_r values. Besides, in Figure 5 the closer to perpendicularity the segments are to the radii of the circumference where the cameras are placed, the higher the r_j values are. Additional statistical measures include an average visibility ratio of 96.98%, a 65.29% of the contours being fully visible, and a minimal visibility ratio of 32.70% (Fig. 3.5(f)). As no previous analyses about visibility and resolution of 2D shapes such as the one we present have been performed, this study may serve as a baseline for further research.

3.6.2 Cost function tests

Now, we evaluate the effectiveness and performance of γ (3.11). Given a 2D object, we analyze how the system is configured in different situations. In order to properly highlight the effects of each individual term in γ, we perform the following simulations over a squared contour shape:



(a) Shoe. $v_r = 100.00\%$

(b) Bat. $v_r = 99.73\%$

(c) Deer. $v_r = 98.07\%$

(d) Lizzard. $v_r = 87.86\%$

(e) Spring. $v_r = 59.16\%$

(f) Device. $v_r = 32.70\%$

Figure 3.5: Shapes from the MPEG-7 data set [85] with their visibility ratios. Segments are colored depending on their maximum resolution r_j with respect to r_{top}, the absolute maximum resolution. Occluded segments are depicted in red (best seen in color and with zoom). The dashed line circumference indicates the distance d from $\mathbf{g}$ where the omnidirectional cameras with unlimited FOV range are placed.

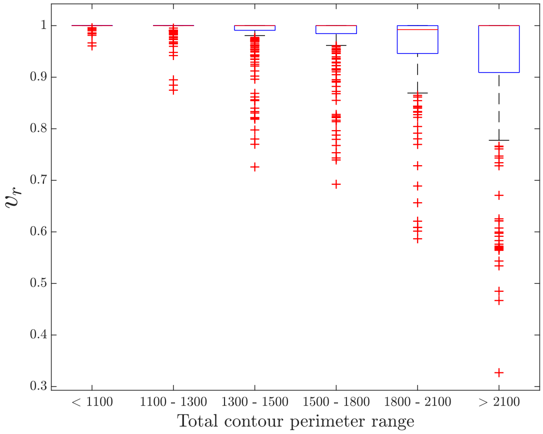


Figure 3.6: Visibility ratios of the 1400 contours from the MPEG-7 data set [85], contained in a single box plot. We can see that as the total perimeter increases, so does scattering, and also that the median is near 1 (full visibility) in all categories.

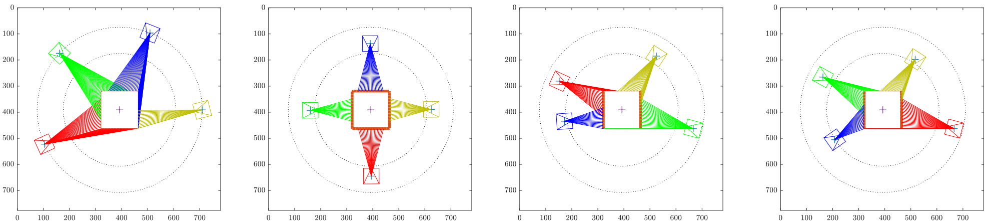
a) $\gamma = \gamma_v$. Only visibility properties are analyzed. $\vartheta_j^* = 1$ for all contour segments, which corresponds to a coverage task where we want the full contour to be inspected by, at least, one camera.

b) $\gamma = \gamma_r$. Only resolution properties are analyzed. $r_j^* = r_j^{max}/2$ for all contour segments, which corresponds to a coverage task where we want the full contour to be detected with, at least, the half of the maximum resolution.

c) $\gamma = (\gamma_v + \gamma_r)/2$. The effects of combining the visibility and resolution terms are studied. $\vartheta_j^* = 1$ for the segments of the top and bottom sides of the square, and $\vartheta_j^* = 0$ for the rest. $r_j^* = r_j^{max}/2$ for the segments of the right and left sides of the square, and $r_j^* = 0$ for the rest. It corresponds to a coverage task where we want some parts of the object to be simply inspected, while others must be inspected more accurately.

d) $\gamma = (\gamma_v + \gamma_r)/2 + \gamma_\sigma$. All terms are considered, including the γ_σ term. The target visibility and resolution are set in the same manner as in statement c). It corresponds to a coverage task similar to the one of the previous case, but here the cameras must be more separated for safety reasons.

For these tests we set $C = 4$ cameras, $S = 100$ segments, $\beta = 30^o$, $L_{max} = 400$ units, and the γ_σ exponent is set to $w = 5$. These cameras are radially constrained to the space between the distances d_{min} and d_{max}, which have been set to 1.5 and 2.2 times the greater side of the contour bounding box, respectively. Figure 3.7 shows the *restricted visibility graphs* optimized at each simulation, and Table 3.1 contains the values each term of γ takes. We can



(a) $\gamma = \gamma_v$ (b) $\gamma = \gamma_r$ (c) $\gamma = (\gamma_v + \gamma_r)/2$ (d) $\gamma = (\gamma_v + \gamma_r)/2 + \gamma_\sigma$

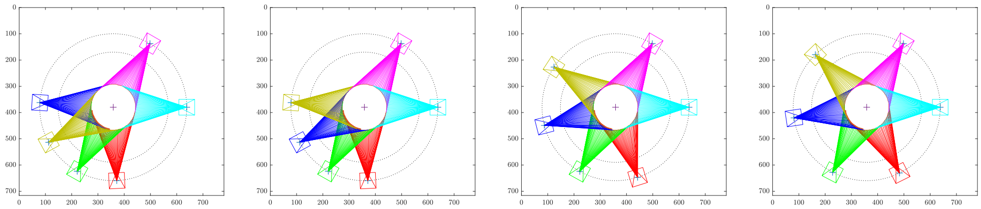
Figure 3.7: Four *restricted visibility graphs* resulting from different test with $C = 4$ cameras, over a squared shape. $\vartheta_j^* = 1$ in the thin orange line segments and $\vartheta_j^* = 0$ in the rest, while $r_j^* = r_j^{max}/2$ in the thick orange line segments and $r_j^* = 0$ in the rest.

see clearly how γ_v induces the cameras to move towards the radial limit d_{max}, so that the FOV is exploited to the highest extent and the maximum number of segments are detected (in fact, in Fig. 3.7(a) the yellow camera could be omitted without affecting the result). A different tendency is driven by γ_r in Fig. 3.7(b), which pushes the cameras to be closer to the object and as perpendicular as possible to the faces of the square, so that higher angular resolutions are obtained. The combined effects of γ_v and γ_r reach the optimized state shown in Fig. 3.7(c), which is modified in Fig. 3.7(d) when adding γ_σ to include wider separations between the cameras.

Table 3.1: Evaluation of the terms of γ with the squared shape, at each one of the four simulations (Fig. 3.7).

Test	F	$F\gamma_v$	$(1-F)\gamma_r$	γ_σ	γ
Fig. 3.7 (a)	1	0	0	8.9956E-5	8.9956E-5
Fig. 3.7 (b)	0	0	9.5917E-1	1.0458E-4	9.5928E-1
Fig. 3.7 (c)	0.5	0	4.7290E-1	1.6684E-2	4.8958E-1
Fig. 3.7 (d)	0.5	0	4.7404E-1	8.1569E-4	4.7485E-1

Additionally to these tests, we study the influence of w in the effects that γ_σ produces. The iterative optimization is applied to a circular contour with $S = 100$ segments, covered by a set of $C = 6$ cameras with FOV restricted to $\beta = 45^o$ and $L_{max} = 800$ units. The radial constraints of the cameras are the same as in the previous case. In this example, we consider a critical section of the contour that must be inspected by at least 4 cameras. Therefore, the target visibility of 20 consecutive segments is set to $\vartheta_j^* = 4$, while for the rest of the segments $\vartheta_j^* = 1$. This configuration produces a dense concentration of cameras. For better understanding the simulation results, the user-defined factor F has been set to 1 to disable the resolution term. Then, there are opposite effects produced by γ_v and γ_σ in such setup: while the cameras tend to be close by the effect of γ_v and the achievement of the required coverage,



(a) $w = 5$ (b) $w = 4$ (c) $w = 3$ (d) $w = 2$

Figure 3.8: Four *restricted visibility graphs* resulting from test cases with different values of w in the γ_σ term, over a circular shape. Thick orange lines represent those segments that must be detected by 4 cameras ($\vartheta_j^* = 4$).

γ_σ forces them to be separated. The balance here is modified by assigning decreasing values to the w exponent. This extends the effective distance of γ_σ, which forces the cameras to be more and more segregated.

Figure 3.8 shows the graphical results of these optimizations for $w = 5, 4, 3, 2$, and Table 3.2 summarizes, for each w, the values the terms in γ take. It can be seen that at the beginning, as w decreases, γ_v remains constant and γ_σ increases. Then, when $w = 3$, γ_v starts to grow and γ_σ decreases, and finally for $w = 2$ both values increase.

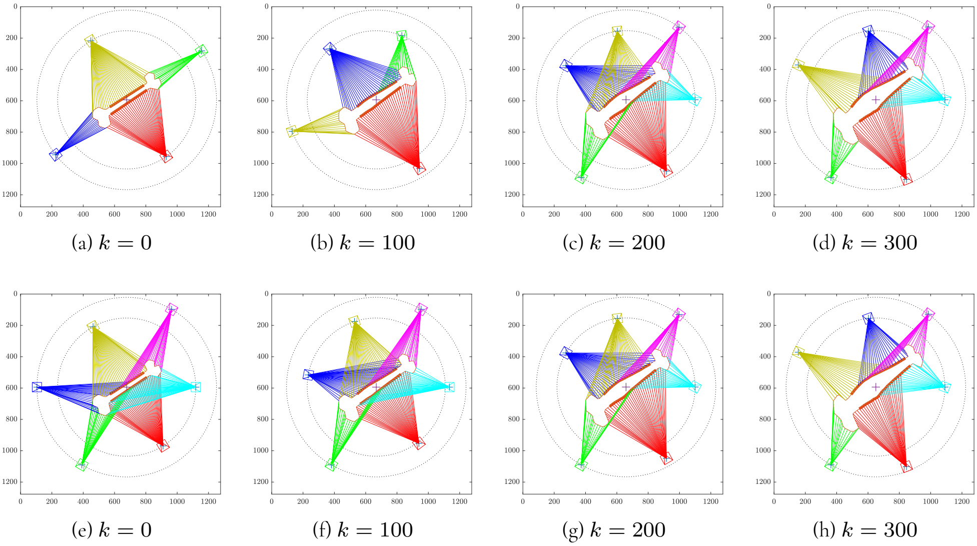
Table 3.2: For each simulation with a different value of w, over the circular shape, values of the terms in γ (Fig. 3.8).

Test	γ_v	γ_σ	γ
$w = 5$	3.7500E-2	1.6720E-2	5.4220E-2
$w = 4$	3.7500E-2	3.8068E-2	7.5568E-2
$w = 3$	1.1875E-1	2.0267E-2	1.3902E-1
$w = 2$	1.5625E-1	5.2453E-2	2.0870E-1

These results show how adaptive the cost function is to different coverage tasks. By adjusting the configuration parameters (namely, F and w) we can give preference to the visibility goal over the resolution goal or vice versa, or favor greater separations between the cameras in case higher levels of safety are needed.

3.6.3 Offline optimization tests

Our offline techniques for solving the minimization problem over a complete deformation process, are evaluated next. In this case, we consider a bone shape taken from the MPEG-7 data set and we sample it in S = 100 segments. We model deformation by applying different exponential distortion functions to the initial shape. Cameras' FOV are here constrained to $\beta = 60^{\circ}$ and $L_{max} = 600$ units, the minimum radial limit d_{min} is equal to the greater side



(a) $k = 0$ (b) $k = 100$ (c) $k = 200$ (d) $k = 300$

(e) $k = 0$ (f) $k = 100$ (g) $k = 200$ (h) $k = 300$

Figure 3.9: From (a) to (d), *restricted visibility graphs* at four different deformation instants, resulting from the iterative optimization over a bone shape. The *restricted visibility graphs* resulting from the optimization with a set of $C^* = 6$ cameras, over the same shape and at the same four deformation instants, are shown from (e) to (h). $\vartheta_j^* = 1$ in the thin orange line segments and $\vartheta_j^* = 0$ in the rest, while $r_j^* = r_j^{max}/2$ in the thick orange line segments and $r_j^* = 0$ in the rest.

of the contour bounding box, and the maximum limit $d_{max} = 1.3\, d_{min}$. With respect to the optimization objectives, a combination of visibility and resolution is established: $r_j^* = r_j^{max}/2$ and $\vartheta_j^* = 0$ for the segments of the top and bottom straight sides of the bone, and $r_j^* = 0$ and $\vartheta_j^* = 1$ for the rest. This definition simulates an inspection task where the top and bottom sections of the bone must be detected more accurately than the rest, but full contour coverage is still required. The coverage task could be applied, for instance, to a 3D printing process where a precise quality control is needed. Then, Algorithm 3.1 is executed to obtain the minimum number of cameras C^* of the deformation process. Afterwards, the system is optimized at each k with the already obtained C^*.

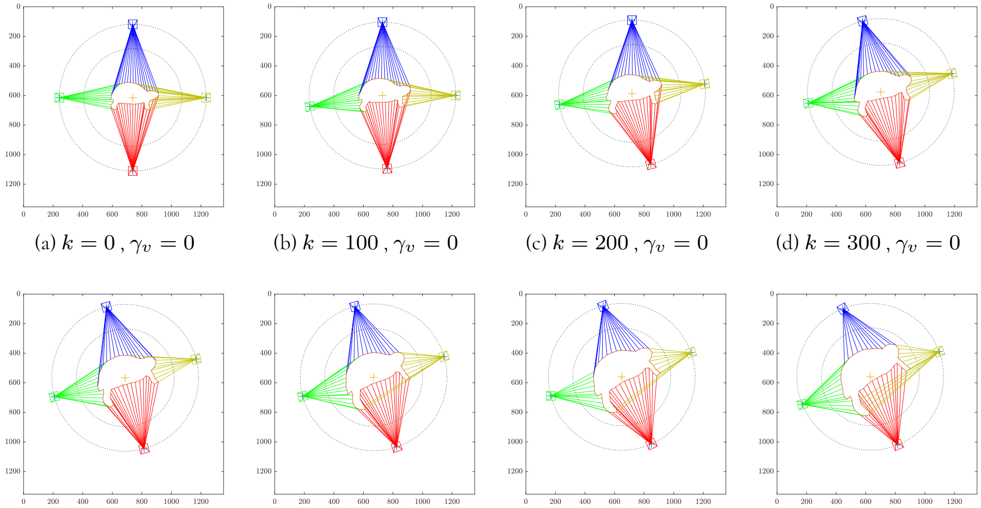
Figure 3.9 shows the *restricted visibility graphs* resulting from the iterative optimization at different k ($k = 0, 100, 200, 300$) from (a) to (d), and from (e) to (h) the *restricted visibility graphs* computed with $C^* = 6$ cameras. Table 3.3 includes, for three different cases we have selected for comparison, the values of each term in γ. The first row of Table 3.3 represents the case where the cameras remain fixed at evenly spaced positions (0, 90, 180 and 270 degrees around $\mathbf{g}(k = 0)$ and $d_i = (d_{max} + d_{min})/2$, $\forall i$), oriented towards $\mathbf{g}(k = 0)$. The second row shows the results of fixing the near-optimal configurations at $k = 0$, and the third one contains the values of each term at the cases shown in Fig. 3.9, from (e) to (h). Even when the system is overpopulated with more cameras than the strictly necessary at some k, it can be seen how the optimized solution with mobile cameras improves both resolution and visibility results of the coverage task. The value of these results lies in the fact that the coverage task is successfully executed by means of a minimal set of sensors. Therefore, we can state that the system of offline near-optimal configured mobile cameras is useful for inspection tasks of deformable objects whose prescribed deformation is approximately known.

Table 3.3: Values of each term in γ, at four deformation instants for comparison of three different setups. The setup of the near-optimal configured mobile cameras corresponds to the cases (e)-(h) in Fig. 3.9.

		k = 0	**k = 100**	**k = 200**	**k = 300**
Evenly spaced fixed cameras (ψ_i : 0°, 90°, 180°, 270°)	γ_v	0	0	8.1818E-2	1.9231E-1
	γ_r	4.8683E-1	4.8559E-1	4.8389E-1	4.8221E-1
	γ_σ	1.1685E-5	1.1685E-5	1.1685E-5	1.1685E-5
	γ	4.8684E-1	4.8560E-1	5.6572E-1	6.7453E-1
Near-optimal configured cameras fixed at $k = 0$	γ_v	0	1.7241E-2	1.1818E-1	1.9231E-1
	γ_r	4.8485E-1	4.8339E-1	4.8151E-1	4.7997E-1
	γ_σ	2.9585E-5	2.9585E-5	2.9585E-5	2.9585E-5
	γ	4.8488E-1	5.0066E-1	5.9972E-1	6.7231E-1
Near-optimal configured mobile cameras	γ_v	0	0	0	0
	γ_r	4.8485E-1	4.8330E-1	4.8130E-1	4.7969E-1
	γ_σ	2.9585E-5	3.8946E-5	4.7513E-5	5.6068E-5
	γ	4.8488E-1	4.8334E-1	4.8135E-1	4.7975E-1

3.6.4 Online optimization tests

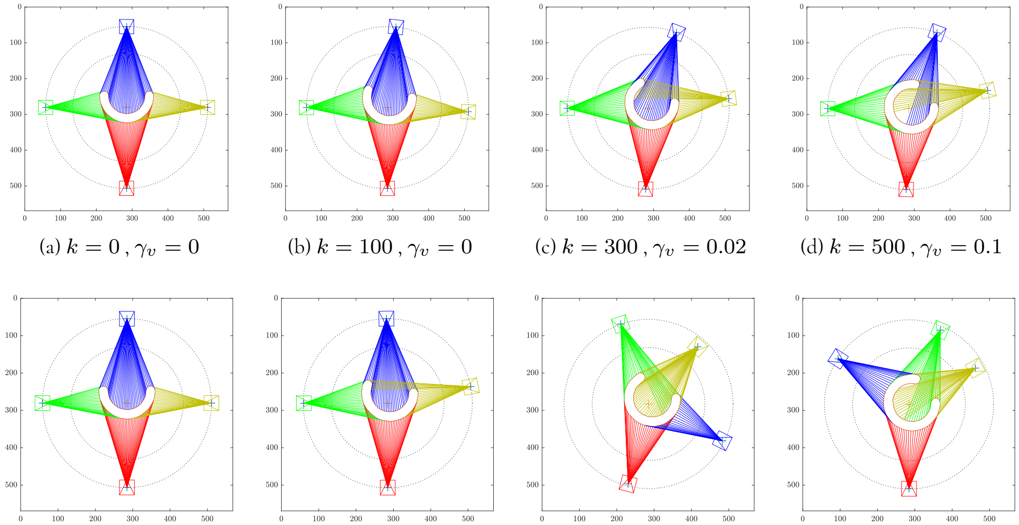
Now we test the local optimization method. In this case, we consider a turtle shape from the MPEG-7 data set, with $S = 50$ segments, increasingly distorted to simulate deformation. Deformation is modeled by applying exponential distortion and a total counter clockwise rotation of 45 degrees. The cameras are modeled with $\beta = 60^{\circ}$ and $L_{max} = 2000$ units, the minimum radial limit d_{min} is set approximately equal to the greater side of the contour bounding box, and the maximum limit $d_{max} = 1.5\, d_{min}$. As for the optimization objectives, the resolution term is omitted for simplicity ($F = 1$), and $\vartheta_j^* = 1$ for all the visible contour segments. We execute the iterative optimization at the first deformation instant, that provides $C^* = 4$ cameras and the initial near-optimal configurations. Afterwards, we apply the local optimization sequentially with each camera at each instant, for a maximum allowed relative difference between $\mathbf{c}_i(k)$ and $\mathbf{c}_i(k+1)$ of 0.05. Figure 3.10 contains the visibility graphs of this test at eight different k ($k = 0, 100, ..., 700$). From (a) to (d) the entire contour is successfully detected, and from (e) to (h) only two segments below the turtle's tail are lost. In comparison with a local optimization, which takes in this case 0.4 seconds on average to solve, the last global optimization with four cameras is 7.5 times slower, taking around 3 seconds. The relevance these results show is related to the achievement of the coverage objectives in a much faster manner than the offline method, which is fast enough so that the approach can reconfigure the system online, and also to the fact that deformation of the object over time is unknown.



(a) $k = 0\,,\, \gamma_v = 0$ (b) $k = 100\,,\, \gamma_v = 0$ (c) $k = 200\,,\, \gamma_v = 0$ (d) $k = 300\,,\, \gamma_v = 0$

(e) $k = 400\,,\, \gamma_v = 0.04$ (f) $k = 500\,,\, \gamma_v = 0.04$ (g) $k = 600\,,\, \gamma_v = 0.04$ (h) $k = 700\,,\, \gamma_v = 0.04$

Figure 3.10: *Restricted visibility graphs* at different deformation instants, with the corresponding values of γ_v. After the iterative optimization at $k = 0$, local optimizations are performed with a set of $C^* = 4$ cameras. $\vartheta_j^* = 1$ and $r_j^* = 0$ for all the visible contour segments.



(a) $k = 0\,,\, \gamma_v = 0$ (b) $k = 100\,,\, \gamma_v = 0$ (c) $k = 300\,,\, \gamma_v = 0.02$ (d) $k = 500\,,\, \gamma_v = 0.1$

(e) $k = 0\,,\, \gamma_v = 0$ (f) $k = 100\,,\, \gamma_v = 0$ (g) $k = 300\,,\, \gamma_v = 0$ (h) $k = 500\,,\, \gamma_v = 0.08$

Figure 3.11: *Restricted visibility graphs* at different deformation instants of a 'C' shape, with the corresponding values of γ_v. $\vartheta_j^* = 1$ and $r_j^* = 0$ for all the visible contour segments. After the global optimizations at $k = 0$ in (a) and (e), with $C = 4$ cameras, the results of applying local ((b)-(d)) and global ((f)-(h)) optimizations are shown for comparison.

3.6.5 Global and local optimizations comparison tests

We perform an additional test in order to compare the offline global and online local optimizations in more detail. A ‘C’ shape, with ends that progressively approach to each other, is sampled with $S = 100$ segments. The FOV of the cameras is set as in the previous test, as well as the radial limits and the optimization objectives. Then, the same deformation sequence is evaluated with the local optimization technique (see Fig. 3.11 (a)-(d)), with a threshold of the 5% of allowed deviation between $\mathbf{c}_i(k)$ and $\mathbf{c}_i(k+1)$, and the global optimization one (see Fig. 3.11 (e)-(h)). Results show that the global optimizations perform slightly better than the local ones, with an improvement of 0.02 in γ_v at the last two deformation instants. However, the computational cost is much higher: a global optimization takes 13.96 seconds on average to solve the system, while a local one takes just 1.04 seconds. Thus, we would select the online strategy for those inspection tasks where the deformation of the object over time is unknown. Otherwise, we would opt for the offline method, which is not as fast as the online one but provides better quality results. It is worth mentioning that the order of the cameras may change from k to $(k+1)$ in the global optimization (e.g., from Fig. 3.11(f) to Fig. 3.11(h) the blue camera changes twice its order into the cameras sequence). This does not represent a problem, due to the fact that the near-optimal configurations are assigned to the cameras of the real-world system so that the total traveled distance is minimized, while the angular order of them is preserved.

3.6.6 Real experiment

Finally, we test the global optimization in a real setup. Figure 3.12 shows an experimental setup where the goal is to perform full visibility coverage with three cameras, of a sole and last assembly task. In this illustrative example, the evaluation of such setup may be performed visually, but numerical analysis of the result cannot be provided accurately due to the complexity of the shapes. Therefore, we propose an additional experimental setup to facilitate the quantitative validation of the coverage system. A 21×21 [cm] paper structure is sequentially folded to simulate a three-steps deformation process (Figure 3.13). This structure is horizontally divided into 28 segments of 3 [cm] in length, which are imprinted and numbered in the faces of the structure. Due to the fact that the object itself is segmented, we can perform the coverage task with a standard color camera whose optical center's position is known. With the segments directly imprinted in the object, we also facilitate the interpretation of the results. In this case, we consider an Intel RealSense D435 module installed on a tripod, with a color camera of 69 ± 1 degrees horizontal FOV ($\beta = 69^o$). We assume that there are no range constraints in this setup ($L_{max} \leftarrow \infty$). The optimization objective is set as $r_j^* = r_j^{max}/4$ and $\vartheta_j^* = 0$ for the segments $\{1, 2, 3, 4, 12, 13, 14, 20, 21, 25, 26, 27, 28\}$ and $r_j^* = 0$ and $\vartheta_j^* = 1$ for the rest, with $F = 0.5$. We execute algorithm 3.1 to obtain the minimum number of cameras C^* of the deformation process, and we obtain $C^* = 3$. Then, the system is optimized with $C = 3$ at every deformation instant, and we get the optimal configuration of the cameras. Figure 3.14 shows the simulation results and the images taken by the Intel RealSense

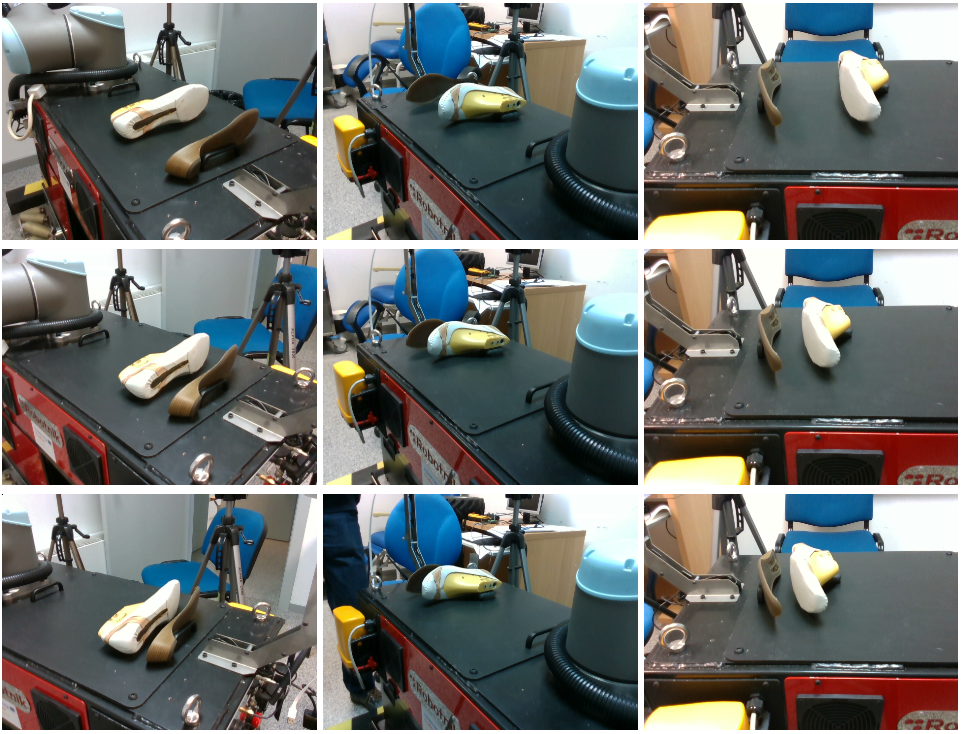


Figure 3.12: Views of three cameras (left to right) in a full visibility coverage experimental setup, where we represent three steps (top to bottom) of a sole-last assembly task.

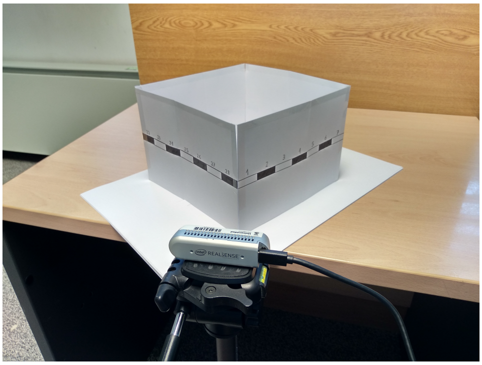
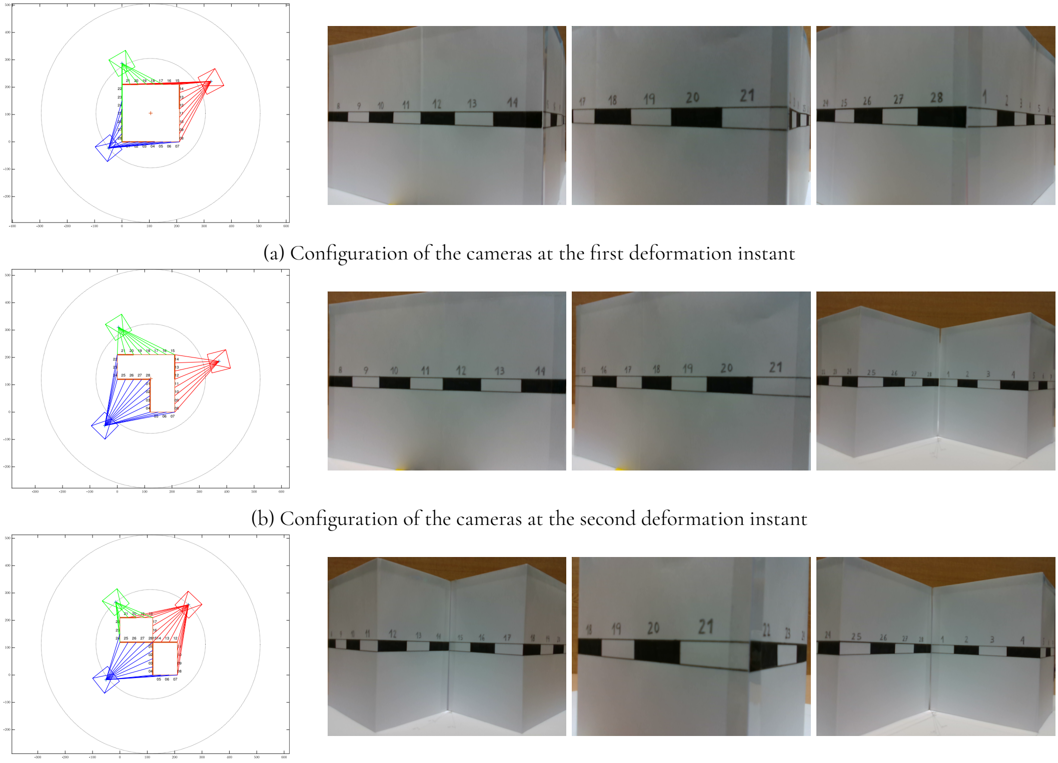


Figure 3.13: View of the real experiment setup. The Intel RealSense D435 module is on a tripod that is positioned and oriented to get the different optimal configurations of the cameras.



(a) Configuration of the cameras at the first deformation instant

(b) Configuration of the cameras at the second deformation instant

(c) Configuration of the cameras at the third deformation instant

Figure 3.14: Real experiment results. At the left, the first images correspond to the *restricted visibility graphs* that result after executing the offline optimization for $C = 3$. The next three images at the right side correspond to the views of the Intel RealSense color camera in the red, green and blue configurations respectively.

color camera from the different optimal configurations at each deformation instant. We can see that all segments are successfully detected, which means the visibility coverage is achieved. Figure 3.15 shows the results concerning the angular resolution of the selected contour segments with $r_j^* \neq 0$ (the resolution of the rest of the segments is not measured). We measure the angular resolution directly from the images, by mapping the horizontal resolution of the segments in pixels to the corresponding section of the FOV (in this case, the horizontal resolution of the images is 640 pixels, which corresponds to the 69 degrees FOV of the color camera). We see that the target resolution is achieved and improved in all segments, which means that the resolution coverage is achieved. These results confirm the validity and utility of the methods we propose for the application in real perception tasks.

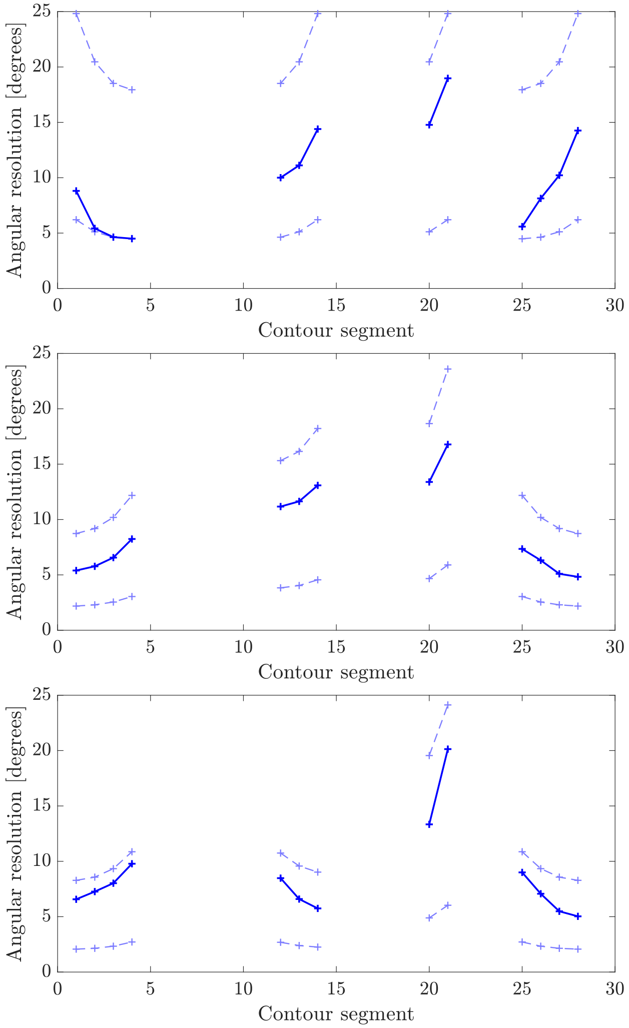


Figure 3.15: From top to bottom, we show the angular resolution results in the first, second and third deformation instants of the real coverage task. The top dashed lines correspond to the maximum angular resolution of the segments, the bottom dashed line to the target resolution and the continuous line to the resolution measured from the images of the Intel RealSense cameras. We can see that the target resolution is achieved and improved in all segments, even getting some results which are close to the maximum values.

3.7 Discussion

In this chapter, we have presented a multi-camera system for coverage of deformable contour shapes, according to a prescribed objective in terms of visibility and resolution. Firstly, we have developed a technique that allows obtaining the theoretical maximum visibility and resolution of a 2D shape at each deformation instant, with an infinite set of omnidirectional and unlimited FOV range cameras. Afterwards, we have exploited this information to define feasible coverage objectives along deformation, and then we have applied an iterative optimization technique that provides the minimum necessary number of cameras for the coverage task and their near-optimal configurations. Therefore, by obtaining a minimal number of cameras we are able to reduce the computational cost of the data processing, and also the synchronization complexity. Finally, we have proposed a method based on local optimizations that outputs near optimal configurations much faster than the global optimization. This method is also applicable online in cases where deformation evolves in an unknown manner. We have tested our approaches in different scenarios, and the results show they are effective and useful for performing coverage tasks in a wide range of cases: for different shapes, deformation processes, coverage objectives, safety conditions and spatial constraints. It is worth noting that our optimization algorithms guarantee the achievement of the target coverage. However, depending on the specific sensors that are considered for the real perception task, the final error of the recovered object shape may vary. In this respect, it is important to properly define the coverage task to obtain the required information of the object.

We go one step further into the problem of multirobot manipulation of deformable objects in the following chapter. After having dealt with deformation from the perception side, we tackle the transport of deformable loads by means of a new deformation model. This model is based on global deformation measurements, of the bounding box that contains the transported object. In turn, the model is exploited in an optimization framework that is constrained with a control barrier function, for achieving the target configuration without collisions.

Chapter 4

Multirobot transport of deformable objects with DBB model

In this chapter we present a multirobot approach for steering a large object to a target configuration, in terms of object dimensions, orientation and position. We present a deformation model based on the evolution over time of the dimensions and rotation of the object's bounding box, for manipulators with double-integrator dynamics. Then, we apply this model in a quadratic programming-based framework, which includes a set of nominal controllers, for achieving the target configuration, and a control barrier function for collision avoidance. We report simulation results to show the performance of this approach in different test scenarios.

4.1 Introduction

PARADIGMATIC manipulation tasks of deformable objects involve their transport and simultaneous control of deformation, with different levels of complexity. This is the case in industries such as the textile or the footwear manufacturing ones, where large fabric parts are transported and progressively transformed between different working stations. Automation of these tasks requires to deal with different challenges, which range from measuring deformation of the object to coordinating the different robots for achieving the manipulation goals [141]. However, despite their complexity, autonomous systems for transport and manipulation of deformable objects could improve the safety of workers and the efficiency of processes. Sectors that could also obtain potential benefits from this approach include construction and logistics. Modern construction techniques incorporate composite materials that are reinforced with layers of fabric, which could be transported and applied by robotic teams. As for the logistics sector, the capacity to handle deformable goods with autonomous systems could increase the storage capacity and would reduce the cost of rigid packaging. In this chapter, we address the problem of transport tasks of large deformable objects that must be steered to a target configuration, while avoiding collisions.

When manipulating deformable objects that are large, fragile, heavy or difficult to grasp, multiple manipulators are usually required [58]. Multirobot systems extend the robustness and resilience of the single agent methods, which are more sensitive to perturbations, but the control actions must be tightly coordinated due to the nature of the task [114]. We refer to robustness as the capability of a system to work under disturbances that do not cause structural changes on the system, and resilience as the ability of a system to autonomously recover its original function from external and/or internal disturbances, which cause an interruption in the normal operation of the system [131]. Works dealing with multirobot transport of deformable objects include a pioneer approach for nonholonomic manipulators based on nonsmooth Lyapunov functions, with guaranteed collision avoidance [128]. In that proposal, deforming the object is the means to avoid obstacles, but it is not a control goal by itself and any feasible deformation state is expected. Another study describes the design of a table with multiple actuators, with caterpillar locomotion-based mechanisms, for manipulation and transport of delicate objects [40]. Simultaneous control of the shape, scale, rotation and position of a formation of robots carrying a deformable object, is achieved with a formation control method [11]. If some specific shape and scale is required, deformation of the object in 2D can be explicitly controlled in that work. A different approach for simultaneous transport and shape control was developed for robots with non-holonomic constraints [91]. Those methods do not tackle the problem of explicitly controlling the deformation of the object in 3D, and they do not consider collision avoidance in their formulation. We can also find approaches based on Dynamic Movement Primitives (DMP) [31] or human-robot collaborative systems for transport of highly deformable planar materials [81], which do not test collision avoidance against dynamic obstacles.

Despite being challenging, modeling deformation is a fundamental step in manipulation of deformable objects. There are many available approaches for this purpose, ranging from the well-known finite element method to the more recent deep learning-based developments [136]. One of the main differences between our proposal and the deep learning methods, which are very popular nowadays, is the required amount of information. Compared to our approach, they require more data for being trained to accomplish the desired task, and it is more difficult to predict the behavior of neural networks. In our work, we choose a geometric model based on a deformable bounding box of the object. Different variants of this concept have been exploited in the past. A 2D version, called *outlined rectangle*, has been considered to obtain a convenient shape of the multirobot transportation system in previous studies [25], [75]. Similarly, the 2D deformable box is adopted in a motion planning strategy for approximating the shape of a team of manipulators and the transported rigid payload [124]. Another study about probabilistic roadmap motion planning for deformable robots obtains the deformation of the robot from a deformed bounding box [23]. A deformable bounding box with variable width is also utilized to describe the shape of an hexapod robot for navigation through confined spaces [22]. None of these previous works obtains a deformable bounding box model in 3D as the one we propose. Our approach allows predicting the evolution of the three dimensions of the bounding box under a specific set of control actions.

Guaranteeing a safe operation under uncertainties and unexpected events constitutes one of the greatest concerns in robotics. Collision avoidance is a classical safety problem, which has been solved with different optimization techniques, such as sequential convex programming [4], [6] and model predictive control (MPC) [21], [124], to name a few. Control barrier functions (CBFs) represent a powerful tool for keeping the state of a system in the safe region. CBFs can be applied to restrict the outputs from the nominal controller of a system, which potentially can induce unsafe behaviors (e.g. collisions). Collision avoidance systems based on this technique have been developed for systems of multiple mobile robots [132] or UAVs [55], [118].

The method we propose builds upon our previous proposal for multirobot transport of deformable objects with minimal motions [62]. In this work, we presented the deformable bounding box (DBB) paradigm, and we described how to obtain a DBB model applying maximization of observability indexes. The navigation task was solved as an optimization problem, in which collision avoidance was achieved with minimal motions of the grippers. In contrast to this former study, the 3D deformable bounding box model is now built in terms of accelerations. Even though the new model has the same structure, the dynamics of the grippers are significantly different with respect to the single integrator alternative. Also in [62] the model was not exploited to have direct control over the shape and orientation around the vertical axis of the object's BB. Another difference is that in the former study the motion of the transported object was constrained to a given route, but in the present work the route is computed online by the QP controller. Additionally, our new collision avoidance system based on the CBF is more robust and effective than the previous system, in which a nonlinear optimization was developed. The new system allows transporting a deformable object in 3D environments to a desired position, without collisions, while simultaneously deforming and rotating it to the desired dimensions and orientation, without overstretching the object during the process. Our main contributions in the present chapter are the deformable bounding box model for mobile manipulators with double-integrator dynamics, the CBF formulation for collision avoidance between the deformable object and static or dynamic obstacles, and the quadratic programming (QP) controller, which integrates the nominal controllers with the CBF and additional safety constraints.

4.2 Problem statement

Let us consider a deformable object to be transported in a 3D environment by $N \geq 3$ mobile robots, which move on the floor following double-integrator dynamics. Each robot mounts a rotational-free gripper that grasps a set of points $\mathbf{p}_i \in \mathbb{R}^3$, $i = 1, ..., N$ in the object boundary (see Fig. 4.1). We model the object shape by its *ground-parallel bounding box*, defined as follows [62]:

Definition 4.1. ***Ground-parallel bounding box.*** *The ground-parallel bounding box (BB) is a box that*

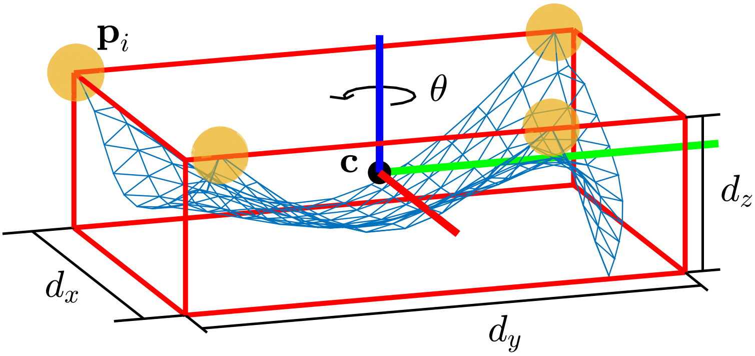


Figure 4.1: Elastic sheet (blue mesh), held by four grippers (large spheres), and its bounding box, formed by the red lines that intersect between them. The main parameters of the bounding box (BB) are: the centroid $\mathbf{c}$, the dimensions d_x, d_y and d_z, and the rotation around the vertical axis θ. A grasping point $\mathbf{p}_i$ and the BB x-y-z reference frame in $\mathbf{c}$ are also depicted.

contains the object and whose top and bottom faces are parallel to the ground plane. It is defined by its centroid $\mathbf{c} \in \mathbb{R}^3$, the box dimensions $\mathbf{d} = [d_x, d_y, d_z]^\intercal$ and the box orientation around the vertical axis $\boldsymbol{\theta} = [0, 0, \theta]^\intercal$. $\mathbf{d}$ and $\boldsymbol{\theta}$ are expressed in the BB reference frame, with origin in $\mathbf{c}$ and whose axes are perpendicular to the BB faces.

This structure is oriented around the main plane of motion in our problem, and therefore considers the rotations with greater impact in the transport task. Note that the rotations around the remaining axes, with much lower impact, are not included as a design criterion to reduce the dimensionality of the problem. However, we will show that 4 degrees of freedom (x, y and z dimensions and rotation around z) add flexibility to avoid obstacles in 3D, while offering a wide range of possible control configurations. We obtain the BB of the object from a set of points $\mathbf{v}_m$, $m = 1, ..., M$ measured on its surface, by means of range sensors or fiducial markers. We do not need these points to be accurate, as long as they represent an approximate shape of the transported object.

We consider that there are obstacles in the environment, described by a set of points $\mathbf{q}_j \in \mathbb{R}^3$, $j = 1, ..., J$ and detected by the system with onboard range sensors, up to a limit distance R.

The task we envision consists in steering the BB of the grasped object without collisions to a target configuration, defined as the combination of position, dimensions, and orientation. This can be understood as a conventional transport process (i.e. translating the object from one place to another) with additional control over the shape and orientation of the object along translation. We propose the following errors to assess the completion of this task:

$$\mathbf{e}_c = \mathbf{c} - \mathbf{c}_t \ , \tag{4.1}$$

$$\mathbf{e}_d = \mathbf{d} - \mathbf{d}_t \ , \tag{4.2}$$

$$e_\theta = \theta - \theta_t \ , \tag{4.3}$$

where $\mathbf{c}_t \in \mathbb{R}^3$ is the target position of the BB centroid, which is set in the 2D plane (the z coordinate is ignored), $\mathbf{d_t} \in \mathbb{R}^3$ are the target dimensions of the BB and θ_t is the target BB orientation around the vertical axis.

4.3 Deformable bounding box model

4.3.1 Model description

In this section we define the object deformation model considered in our study. The model is based on the *deformable bounding box* paradigm, which extends the concept of BB (Definition 4.1) with new properties.

Definition 4.2. ***Deformable bounding box.*** *The deformable bounding box (DBB) is the BB whose dimensions and rotation change over time when certain actions are applied to the object. These actions are traction/compression accelerations in the horizontal plane* $\ddot{\mathbf{G}} = [\ddot{G}_x, \ddot{G}_y, 0]^\intercal$ *and vertical rotation acceleration* $\ddot{\boldsymbol{\phi}} = [0, 0, \ddot{\phi}]^\intercal$. *As the object deforms, the DBB evolves smoothly between consecutive time instants.*

The process for obtaining this kind of BB starts by computing the ground-parallel minimal bounding box of the object. The ground-parallel minimal bounding box is equivalent to the extrusion of the 2D minimal bounding box of the object along the object's height. Then, we obtain the BB reference frame with origin in $\mathbf{c}$ whose axes are perpendicular to the BB faces. Note that the bounding box this model requires does not correspond to the ground-parallel minimal bounding box, whose dimensions and orientation may present sharp changes over time. Therefore, the next step involves applying Principal Component Analysis (PCA) to the collection of vectors $\mathbf{v}_{bm} = \mathbf{v}_m - \mathbf{c}$, $\forall m$. These vectors are stacked in the matrix $\mathbf{V}_b \in \mathbb{R}^{M \times 3}$, which after applying the PCA yields $\mathbf{V}_b = \mathbf{USV}^\intercal$. We can obtain the principal direction in the horizontal plane of the points in the global reference system as $\varphi = \text{atan2}(V_{2,1},\, V_{1,1})$, where the subscripts denote the corresponding elements of the matrix. In this case, the principal direction of the points in 2D constitutes a line that minimizes the distance to them. In the following deformation instants the angles between the principal direction and the horizontal axes of the BB reference will be preserved, which means that the orientation of the BB around the vertical axis will follow the evolution of the principal direction of the object in the horizontal plane. The principal direction represents a convenient reference for orienting the BB, due to the fact that it varies as smoothly as the object points do. Then, the orientation of the BB can be computed as $\theta_k = \theta_0 - \varphi_0 + \varphi_k$. Finally, the dimensions $\mathbf{d}$ of the BB are computed from the maximum and minimum coordinates of the points $\mathbf{v}_m$ in the new BB reference frame.

We build the object deformation model upon Definition 4.2 in the following manner:

$$\left[\ddot{\mathbf{d}}^{\intercal}, \ddot{\theta}\right]^{\intercal} = \mathbf{J}\left[\ddot{G}_x, \ddot{G}_y, \ddot{\phi}\right]^{\intercal} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \\ k_{41} & k_{42} & k_{43} \end{bmatrix} \begin{bmatrix} \ddot{G}_x \\ \ddot{G}_y \\ \ddot{\phi} \end{bmatrix}, \tag{4.4}$$

where

$$\ddot{\mathbf{d}}_k = \left(\dot{\mathbf{d}}_k - \dot{\mathbf{d}}_{(k-1)}\right) / \ \Delta t \ , \tag{4.5}$$

$$\dot{\mathbf{d}}_k = \left(\mathbf{d}_k - \mathbf{d}_{(k-1)}\right) / \ \Delta t \ , \tag{4.6}$$

$$\ddot{\theta}_k = \left(\dot{\theta}_k - \dot{\theta}_{(k-1)}\right) / \ \Delta t \ , \tag{4.7}$$

$$\dot{\theta}_k = \left(\theta_k - \theta_{(k-1)}\right) / \ \Delta t \ . \tag{4.8}$$

$\Delta t = t_k - t_{(k-1)}$ is the time interval between instants k and $k-1$. The interaction matrix $\mathbf{J}$ in (4.4) maps the input accelerations to the deformation $\ddot{\mathbf{d}}$ and rotation $\ddot{\theta}$ accelerations of the DBB by means of 12 parameters. Nine of them vary depending on the object's mechanical properties and the grippers' configuration. We consider that the rotation accelerations are small, and therefore centrifugal effects can be neglected. This makes DBB deformations independent from DBB rotations, i.e. $k_{13} = k_{23} = k_{33} = 0$. The intuition behind this proposal is that the dynamics of deformation can be approximated with sufficient accuracy by this linear model. Given that the dimensions of the BB and, therefore, the dimensions of the object are constrained, as we describe in Section 4.4.3 the set of deformation states is bounded.

Once we have described the main aspects of the model, we need to connect it with the actuators that produce the required inputs. Next equations show how the grippers accelerations in the global reference are computed from $\ddot{G}_x$, $\ddot{G}_y$ and $\ddot{\phi}$:

$$\begin{aligned} \ddot{\mathbf{p}}_i = \mathbf{R}\, \Big(& \ddot{\mathbf{c}}_t^{BB} + \ddot{\boldsymbol{\phi}} \times \mathbf{p}_i^{BB} + \dot{\boldsymbol{\phi}} \times (\dot{\boldsymbol{\phi}} \times \mathbf{p}_i^{BB}) \\ & + 2\dot{\boldsymbol{\phi}} \times \dot{\mathbf{p}}_i^{BB} + \frac{\ddot{\mathbf{G}} \circ \mathrm{sgn}(\mathbf{p}_i^{BB})}{2} \Big) \ , \end{aligned} \tag{4.9}$$

$$\mathbf{p}_i^{BB} = \mathbf{R}^{\intercal}(\mathbf{p}_i - \mathbf{c}) \ , \tag{4.10}$$

$$\mathbf{R} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} , \tag{4.11}$$

where $\ddot{\mathbf{c}}_t^{BB} \in \mathbb{R}^3$ is the target horizontal acceleration of the BB centroid in the BB reference ($\ddot{c}_{tz}^{BB} = 0$, by convention), $\mathbf{R} \in SO(3)$ is the rotation matrix from the BB to the global coordinate systems, '$\times$' is the cross product operator, '$\circ$' is the element-wise product operator and

$\mathbf{p}_i^{BB}$ is the position of gripper i in the BB reference. Equation (4.9) represents the acceleration of the grippers as if they were connected to a rigid BB (see [30, eq. (6.10)] about the linear acceleration of a manipulator according to the rigid body dynamics). However, the last term corrects the resulting acceleration with the deformation inputs to the DBB [62]. The main idea behind this correction term is to preserve the *action-counteraction balance* of the system (see Remark 4.1). This balance allows preventing global displacements of the BB, when deforming it, by applying the half of the traction/compression inputs from opposite directions, at each of the two horizontal axes of the BB reference frame.

One of the most prominent advantages of this model, in contrast with the 2D counterpart, is that it does not only allow to predict horizontal deformations and rotation around z from the inputs $\ddot{\mathbf{G}}$ and $\ddot{\phi}$, but also the vertical deformation of the object's BB. Besides allowing to achieve specific configurations of the BB in 3D, this kind of deformation control can be useful for collision avoidance. For instance, if an obstacle in height is encountered and the imposed constraints are satisfied, the system can surmount the obstacle instead of going around it, and reduce the traveled distance. However, in case a more conservative behavior is sought, in which the bounding box never goes over the incoming obstacles, the user can increase the safety distance or constrain the minimum vertical dimension as we will explain later (see Sections 4.4.2 and 4.4.3).

Remark 4.1. *Note that at least one positive and one negative component of the element-wise multiplication* $\ddot{\mathbf{G}} \circ \operatorname{sgn}(\mathbf{p}_i^{BB})$ *must exist for the two non-zero components of* $\ddot{\mathbf{G}}$. *If this condition is satisfied, the action-counteraction balance is preserved for each BB horizontal axis.*

Next proposition provides the general rule for guaranteeing the action-counteraction balance as described in Remark 4.1.

Proposition 4.1. *Consider a set of* $N \geq 3$ *grippers carrying a deformable object. These grippers are positioned so that they are separated less than* π *radians from each of the neighboring ones, around the object's BB centroid* $\mathbf{c}$. *With this setup, the action-counteraction requirement is met at each BB horizontal axis for any possible orientation of the BB reference frame and the rotated inputs.*

Proof. According to Remark 4.1, the positive and negative contributions of the non-zero components of $\ddot{\mathbf{G}}$ must be performed by, at least, one gripper each. If the grippers are separated less than π radians from each of the neighboring ones, three quadrants of every possible BB reference frame will always include a gripper. Therefore, the action-counteraction requirement will always be met. □

4.3.2 Model identification

In this section, we describe the model identification problem in terms of accelerations, with additional details and further analysis with respect to our previous work [62]. As we mentioned in the previous section, 9 parameters of the DBB model vary according to the particular object that is transported, the number of grippers and the relative position of the grippers with respect to the object. This implies that they must be obtained for each case. The most basic identification strategy consists in taking 3 different measurements of $\ddot{\mathbf{d}}$ and $\ddot{\theta}$ from randomly chosen input actions $\ddot{\mathbf{G}}$ and $\ddot{\boldsymbol{\phi}}$. Then, an estimate of the model parameters would be obtained by solving the system of equations. However, this method is highly sensitive to errors coming from the sensing devices and the specificity of the measurements set, and in general produces low quality results. An ordinary least-squares approach improves the quality of the previous solution by linearly adjusting the parameters from a set of $S > 3$ measurements:

$$[k_{11}, k_{12}, ..., k_{43}]^{\mathsf{T}} = (\mathbf{A}^{\mathsf{T}}\,\mathbf{A})^{-1}\mathbf{A}^{\mathsf{T}}\,\mathbf{b}\ , \tag{4.12}$$

$$\mathbf{A} = \text{blkdiag}(\begin{bmatrix} \ddot{G}_{x1} & \ddot{G}_{y1} \\ \vdots & \\ \ddot{G}_{xS} & \ddot{G}_{yS} \end{bmatrix}, \begin{bmatrix} \ddot{G}_{x1} & \ddot{G}_{y1} \\ \vdots & \\ \ddot{G}_{xS} & \ddot{G}_{yS} \end{bmatrix}, \begin{bmatrix} \ddot{G}_{x1} & \ddot{G}_{y1} \\ \vdots & \\ \ddot{G}_{xS} & \ddot{G}_{yS} \end{bmatrix}, \begin{bmatrix} \ddot{G}_{x1} & \ddot{G}_{y1} & \ddot{\phi}_1 \\ \vdots & & \\ \ddot{G}_{xS} & \ddot{G}_{yS} & \ddot{\phi}_S \end{bmatrix})\ , \tag{4.13}$$

$$\mathbf{b} = [\ddot{d}_{x1}, \ddot{d}_{x2}, ..., \ddot{d}_{xS}, \ddot{d}_{y1}, ..., \ddot{d}_{yS}, \ddot{d}_{z1}, ..., \ddot{d}_{zS}, \ddot{\theta}_1, ..., \ddot{\theta}_S]^{\mathsf{T}}\ , \tag{4.14}$$

with $\mathbf{A} \in \mathbb{R}^{4S\times 9}$ and $\mathbf{b} \in \mathbb{R}^{4S}$.

Although this technique is able to diminish the errors coming from sensor noise and local effects, created by some measurements, the particular choice of measurements and its quantity still affect the accuracy of the resulting DBB model. We obtain a convenient selection of measurements by applying the *observability index maximization* technique. Observability indexes derive from the alphabet optimalities, and they provide statistical information about the numerical conditioning of the model and its variance [66]. The field of robot calibration covers most of the approaches that have exploited the observability index maximization, for purposes like improving the robustness of calibration to sensor noise [33] or for selecting the index that yields the most accurate robot calibrations [78], among other applications.

Let us denote $\sigma_r \leq \sigma_{r-1} \leq ... \leq \sigma_1$ the r singular values of the regressor matrix $\mathbf{A}$. The five observability indexes we study are computed as [78]: $O_1 = (\sigma_r\,\sigma_{r-1}\,...\,\sigma_1)^{1/r}/S^{1/2}$, $O_2 = \sigma_r/\sigma_1$, $O_3 = \sigma_r$, $O_4 = \sigma_r^2/\sigma_1$ and $O_5 = (\sum_{i=1}^{r} 1/\sigma_i)^{-1}$. O_1 represents the volume of a hyperellipsoid in which the singular values correspond to the length of the axes. It is an indicator of the data scatter, i.e. the exploration of the parametric space. Instead of measuring the size of the hyperellipsoid, O_2 quantifies the ratio between the shortest and the largest axes. In other words, it shows how well explored some of the parameters are in comparison with the rest. O_3 indicates the least explored parameter (worst case). O_4 is the combination of O_2 and O_3, and O_5 is the harmonic mean of the singular values divided by the number of model parameters.

In order to maximize these indexes we apply Algorithm 4.2, which is based in the DETMAX algorithm [96]. First, we generate a set of candidate inputs η with different combinations of $\ddot{G}_x$, $\ddot{G}_y$ and $\ddot{\phi}$. Starting with an initial experiment design of S random input combinations from η, the algorithm maximizes an observability index by continuously exchanging inputs (rows of $\mathbf{A}$) between the experiment design and η. The process stops when the index converges to a maximum value, up to a user-defined tolerance. Then, we obtain an optimal matrix $\mathbf{A}$ and, by applying the input combinations it contains to the system, the matrix of measurements $\mathbf{b}$. Finally, we compute from (4.12) the parameters of the DBB model. In contrast to Mitchell's proposal, our algorithm considers a constant experiment size, and instead of obtaining D-optimal experiment designs we optimize the design by maximizing an observability index.

Algorithm 4.2 Maximize an observability index O.

```
$O_{\text{diff}} \leftarrow \infty$
$O_{-1} \leftarrow 0$
optimStop $\leftarrow tolerance$
**exp** $\leftarrow S$ random inputs from $\eta$
**while** $O_{\text{diff}} >$ optimStop **do**
    $O_{\text{prev}} \leftarrow 0$
    **for** $i = 1, ..., \text{size}(\eta)$ **do**
        **newExp** $\leftarrow [\mathbf{exp}, \eta_i]$
        $O \leftarrow computeIndex(\mathbf{newExp})$
        **if** $O_{\text{prev}} < O$ **then**
            $\kappa \leftarrow i$ # *auxiliary variable*
            $O_{\text{prev}} \leftarrow O$
        **end if**
    **end for**
    **exp** $\leftarrow [\mathbf{exp}, \eta_\kappa]$ # *add input that maximizes O*
    **for** $j = 1, ..., S + 1$ **do**
        **newExp** $\leftarrow \left[\text{exp}_1, ..., \text{exp}_{j-1}, \text{exp}_{j+1}, ..., \text{exp}_{S+1}\right]$
        $O \leftarrow computeIndex(\mathbf{newExp})$
        **if** $O_{\text{prev}} > O$ **then**
            $\kappa \leftarrow j$
            $O_{\text{prev}} \leftarrow O$
        **end if**
    **end for**
    **exp** $\leftarrow \left[\text{exp}_1, ..., \text{exp}_{\kappa-1}, \text{exp}_{\kappa+1}, ..., \text{exp}_{S+1}\right]$ # *remove input that less diminishes O*
    $O \leftarrow computeIndex(\mathbf{exp})$
    $O_{\text{diff}} \leftarrow (O - O_{-1})$
    $O_{-1} \leftarrow O$
**end while**
**return** $O$, **exp**
```

This technique allows obtaining suitable DBB models from a reduced set of measurements, which we select in advance to the model identification and the task execution. A comparative analysis of the different observability indexes, in terms of utility and performance, is shown in Section 4.5. In that section we also study how the experiment size S affects the DBB model accuracy, and we justify the selection of O_1 for obtaining the most appropriate DBB models. In the studied test cases, we have realized that a DBB model with constant parameters, identified before performing the task, performs well the transport objectives. Even if not considered here, it could also be interesting to study the effects of updating the model along the manipulation.

4.4 Controller for deformable object transport

4.4.1 Nominal control system

The control actions must steer the system to the desired configuration, reducing to zero the errors $\mathbf{e}_c$ (4.1), $\mathbf{e}_d$ (4.2) and e_θ (4.3). We propose the following set of nominal controllers for mobile manipulators with double-integrator dynamics:

$$\ddot{\mathbf{G}} = -\, k_{1d}\, \mathbf{e}_d - k_{2d}\, \dot{\mathbf{d}}\ , \tag{4.15}$$

$$\ddot{\phi} = -\, k_{1r}\, e_\theta - k_{2r}\, \dot{\theta}\ , \tag{4.16}$$

$$\ddot{\mathbf{c}}_t^{BB} = \mathbf{R}^\intercal(-k_{1t}\, \mathbf{e}_c - k_{2t}\, \dot{\mathbf{c}})\ , \tag{4.17}$$

where k_{1d}, k_{2d}, k_{1r}, k_{2r}, k_{1t} and k_{2t} are control gains. We aggregate the nominal control inputs in a single column vector

$$\mathbf{u}_n = [\ddot{G}_x, \ddot{G}_y, \ddot{\phi}, \ddot{c}_{tx}^{BB}, \ddot{c}_{ty}^{BB}]^\intercal\ . \tag{4.18}$$

Note that these controllers are centralized in order to drive the system with tightly coordinated actions, required in manipulation tasks as the ones we tackle.

4.4.2 Obstacle avoidance with control barrier funcion

Since collision avoidance is not explicitly accounted by the nominal controllers, we introduce a flexible, minimally invasive and robust solution to prevent collisions with obstacles in the environment. In particular, we consider an optimization method based on a CBF, which regulates the nominal control inputs so that the system is always in a safe state. We adopt a modified version of the centralized approach proposed by Wang *et al.* [132], where multiple robots with intersecting trajectories are able to reach target positions while avoiding robot-to-robot and robot-to-obstacle collisions. In our case, the transported deformable object makes collision avoidance more challenging to be guaranteed, as we must also consider object-to-obstacle collisions. Besides, we do not deal explicitly with robot-to-robot collisions, since they are implicitly avoided by constraining the minimum BB dimensions, as we will describe in the next section.

Firstly, we define a set of virtual points $\mathbf{p}_{vl} \in \mathbb{R}^3$, $l = 1, ..., L$ uniformly distributed over the faces of the BB. The dynamics of these points are linked to the DBB as

$$\begin{aligned}\ddot{\mathbf{p}}_{vl} = \ddot{\mathbf{c}}_t^{BB} + \ddot{\boldsymbol{\theta}} \times \mathbf{p}_{vl} + \dot{\boldsymbol{\theta}} \times (\dot{\boldsymbol{\theta}} \times \mathbf{p}_{vl}) + 2\dot{\boldsymbol{\theta}} \times \dot{\mathbf{p}}_{vl} \\ + (\mathbf{p}_{vl} - [0, 0, d_z/2]^\intercal) \circ \ddot{\mathbf{d}} \oslash \mathbf{d} \,.\end{aligned} \tag{4.19}$$

where '$\oslash$' is the element-wise division. Note that the top face of the BB is constant in height, due to the fact that the grippers move in the 2D plane. This is the reason why we include the $-[0, 0, d_z/2]^\intercal$ term, that shifts the zero vertical movement plane to the top face of the BB. Then, we can write an affine system

$$\dot{\mathbf{x}}_l = f(\mathbf{x}_l) + g(\mathbf{x}_l)\mathbf{u} \,, \tag{4.20}$$

where $\mathbf{x}_l = [\mathbf{p}_{vl}, \dot{\mathbf{p}}_{vl}]^\intercal$ and $f(\mathbf{x}_l)$ and $g(\mathbf{x}_l)$ are locally Lipschitz continuous functions. We propose the following criterion for collision avoidance of the system (4.20):

$$\|\mathbf{p}_{vlj}\| + \int_{t_0}^{t_f} \dot{p}_{vlj}^{\perp}(t)dt \geq D_{lj}^{min} \,, \tag{4.21}$$

where $\mathbf{p}_{vlj} = \mathbf{p}_{vl} - \mathbf{q}_j^{BB}$, $\mathbf{q}_j^{BB} = \mathbf{R}^\intercal(\mathbf{q}_j - \mathbf{c})$,

$$\dot{p}_{vlj}^{\perp} = \frac{\mathbf{p}_{vlj}^\intercal}{\|\mathbf{p}_{vlj}\|}\dot{\mathbf{p}}_{vlj} \tag{4.22}$$

is the relative velocity between $\mathbf{p}_{vl}$ and $\mathbf{q}_j^{BB}$ projected in the normal direction, $\dot{\mathbf{p}}_{vlj} = \dot{\mathbf{p}}_{vl} - \dot{\mathbf{q}}_j^{BB}$, $\dot{\mathbf{q}}_j^{BB} = \mathbf{R}^\intercal(\dot{\mathbf{q}}_j - \dot{\mathbf{c}})$ and

$$t_f = \frac{\dot{p}_{vlj}^{\perp}(t_f) - \dot{p}_{vlj}^{\perp}(t_0)}{\alpha_l} + t_0 \,. \tag{4.23}$$

At time t_f the points completely stop ($\dot{p}_{vlj}^{\perp}(t_f) = 0$) when applying the maximum deceleration $\alpha_l = \|\ddot{\mathbf{p}}_{vl}\|_\infty$, and D_{lj}^{min} is the minimum allowed distance between the virtual point $\mathbf{p}_{vl}$ and the obstacle $\mathbf{q}_j^{BB}$. Note that we take equations (4.20), (4.21), (4.22) and (4.23) as proposed by Wang *et al.* [132]. By substituting (4.23) and (4.22) in (4.21) and solving the definite integral, we obtain the candidate CBF

$$h_{lj} = 2\,\alpha_l\,(\|\mathbf{p}_{vlj}\| - D_{lj}^{min}) - \dot{p}_{vlj}^{\perp 2} \,. \tag{4.24}$$

According to its definition [7], h_{lj} is a suitable CBF if the following expression is satisfied:

$$\sup_{\mathbf{u}}[\mathcal{L}_f h_{lj}(\mathbf{x}_l) + \mathcal{L}_g h_{lj}(\mathbf{x}_l)\mathbf{u}] \geq -\varepsilon(h_{lj}(\mathbf{x}_l)) \,, \tag{4.25}$$

where $\mathcal{L}$ stands for the Lie derivative, $\mathbf{u} \in \mathbb{R}^5$ similarly to (4.18) is the control input and ε is an extended class K_∞ function. Then, we apply this definition to our candidate CBF to obtain the set of linear constraints

$$\begin{aligned}
\mathbf{p}_{vlj}^{\mathsf{T}}\dot{\mathbf{p}}_{vlj}\frac{\mathbf{p}_{vlj}^{\mathsf{T}}}{\|\mathbf{p}_{vlj}\|}\mathbf{d}_M\mathbf{J}_{ext}\mathbf{u} \leq{}& \frac{\|\mathbf{p}_{vlj}\|}{2}\varepsilon(h_{lj}(\mathbf{x}_l)) + \alpha_l\mathbf{p}_{vlj}^{\mathsf{T}}\dot{\mathbf{p}}_{vl} \\
&- \mathbf{p}_{vlj}^{\mathsf{T}}\dot{\mathbf{p}}_{vlj}\frac{\mathbf{p}_{vlj}^{\mathsf{T}}}{\|\mathbf{p}_{vlj}\|}(\boldsymbol{\theta}_M\mathbf{p}_{vl} + \boldsymbol{\theta}_{MV}\dot{\mathbf{p}}_{vl}) \,, \qquad (4.26)
\end{aligned}$$

where

$$\mathbf{d}_M = \begin{bmatrix} \frac{p_{vlx}}{d_x} & 0 & 0 & -p_{vlx} & 1 & 0 \\ 0 & \frac{p_{vly}}{d_y} & 0 & p_{vlx} & 0 & 1 \\ 0 & 0 & \frac{p_{vlz}}{d_z} - 0.5 & 0 & 0 & 0 \end{bmatrix} , \qquad (4.27)$$

$$\mathbf{J}_{ext} = \begin{bmatrix} \mathbf{J} & \mathbf{0}_{4\times 2} \\ \mathbf{0}_{2\times 3} & \mathbf{I}_{2\times 2} \end{bmatrix} , \qquad (4.28)$$

$$\boldsymbol{\theta}_M = \left[-\dot{\theta}^2, -\dot{\theta}^2, 0\right] \mathbf{I}_{3\times 3} \,, \qquad (4.29)$$

$$\boldsymbol{\theta}_{MV} = \begin{bmatrix} 0 & -2\dot{\theta} & 0 \\ 2\dot{\theta} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} . \qquad (4.30)$$

These equations constraint the system to maintain a safety distance D_{lj}^{min} with every obstacle point $\mathbf{q}_j$. Therefore, obstacles are treated as spheres of radius D_{lj}^{min}, but only for collision avoidance. The function ε determines how close the system will remain to the safety boundary when avoiding collisions.

4.4.3 Quadratic programming controller

The nominal control inputs $\mathbf{u}_n$ must be modified to avoid collisions when obstacles intersect with the desired route. We achieve the collision-free configurations by means of a quadratic programming-based (QP) controller, which includes the CBF:

$$\begin{aligned}
&\text{Given} \quad \mathbf{u}_n, \mathbf{p}_{vl}, \dot{\mathbf{p}}_{vl}, \mathbf{q}_j^{BB}, \mathbf{J}, \mathbf{d}, \dot{\boldsymbol{\theta}}, D_{lj}^{min}, \alpha_l, \forall l, j \\
&\underset{\mathbf{u}}{\text{minimize}} \quad \xi = \sum_{i=1}^{5} k_{wi}(u_i - u_{ni})^2 \qquad (4.31) \\
&\text{subject to:} \\
&\qquad \mathbf{A}_{lj}\mathbf{u} \leq b_{lj} \;, \forall l, j, \; l = 1, ..., L, \; j = 1, ..., J \;, \\
&\qquad \|\mathbf{u}\|_\infty \leq \alpha_l, i = 1, ..., N
\end{aligned}$$

where $k_{wi} > 0, \forall i$ are control weights that satisfy $\sum_{i=1}^{5} k_{wi} = 1$, L is the number of virtual points on the BB, J is the number of points describing the obstacles,

$$\mathbf{A}_{lj} = \mathbf{p}_{vlj}^{\mathsf{T}} \dot{\mathbf{p}}_{vlj} \frac{\mathbf{p}_{vlj}^{\mathsf{T}}}{\|\mathbf{p}_{vlj}\|} \mathbf{d}_M \mathbf{J}_{ext} \ , \tag{4.32}$$

$$\begin{aligned} b_{lj} =& \frac{\|\mathbf{p}_{vlj}\|}{2} \varepsilon(h_{lj}(\mathbf{x}_l)) + \alpha_l \, \mathbf{p}_{vlj}^{\mathsf{T}} \dot{\mathbf{p}}_{vl} \\ & - \mathbf{p}_{vlj}^{\mathsf{T}} \dot{\mathbf{p}}_{vlj} \frac{\mathbf{p}_{vlj}^{\mathsf{T}}}{\|\mathbf{p}_{vlj}\|} (\boldsymbol{\theta}_M \mathbf{p}_{vl} + \boldsymbol{\theta}_{MV} \dot{\mathbf{p}}_{vl}) \ . \end{aligned} \tag{4.33}$$

The k_{wi} constants allow regulating the effects of the QP controller over the nominal controller. For instance, if we take higher values of k_{w1} and k_{w2} than those of k_{w3}, k_{w4} and k_{w5}, nominal rotations and global displacements will have preference over nominal deformations for being modified.

We can also implement additional constraints in this structure to avoid overstretching the object and also to limit the rotation and translation velocities. If we assume uniform accelerations between instants k and $k+1$, the linear inequalities

$$\begin{bmatrix} 2\Big((\mathbf{d}^{max} - \mathbf{d}_k)/\Delta t^2 - \dot{\mathbf{d}}_k/\Delta t\Big) \\ (\dot{\theta}^{max} - \dot{\theta}_k)/\Delta t \\ (\dot{\mathbf{c}}^{max} - [\dot{c}_{kx}, \dot{c}_{ky}]^{\mathsf{T}})/\Delta t \end{bmatrix} \geq \mathbf{J}_{ext} \, \mathbf{u} \ , \tag{4.34}$$

$$\begin{bmatrix} 2\Big((\mathbf{d}^{min} - \mathbf{d}_k)/\Delta t^2 - \dot{\mathbf{d}}_k/\Delta t\Big) \\ (\dot{\theta}^{min} - \dot{\theta}_k)/\Delta t \\ (\dot{\mathbf{c}}^{min} - [\dot{c}_{kx}, \dot{c}_{ky}]^{\mathsf{T}})/\Delta t \end{bmatrix} \leq \mathbf{J}_{ext} \, \mathbf{u} \ , \tag{4.35}$$

guarantee that $\mathbf{d}$, $\dot{\theta}$ and $\dot{\mathbf{c}}$ will be bounded between a set of maximum ($\mathbf{d}^{max}$, $\dot{\theta}^{max}$, $\dot{\mathbf{c}}^{max}$) and minimum ($\mathbf{d}^{min}$, $\dot{\theta}^{min}$, $\dot{\mathbf{c}}^{min}$) defined limits. Note that the constraint over the BB vertical dimension can also prevent the object from touching the ground, since the grippers travel at constant height.

By means of this controller, we obtain the closest control input to the nominal one that satisfies the constraints from the CBF and the linear inequalities (4.34) and (4.35). The proposed centralized formulation induces tightly coordinated motions of the robots that aim at preserving the deformable object without damages. In addition, we consider a small number of mobile manipulators in the studied practical cases. Therefore, the centralized version of the CBF and the QP controller seems appropriate for the current state of the method.

4.5 Results

4.5.1 Observability indexes analysis

In this section, we will identify the observability index that provides the most appropriate DBB model from a reduced set of measurements, for different grippers configurations and materials. We evaluate the quality of a model from its mean squared error, which is computed as

$$MSE = \frac{1}{S}\sum_{i=1}^{S}([\ddot{\mathbf{d}}_i, \ddot{\theta}_i] - [\tilde{\ddot{\mathbf{d}}}_i, \tilde{\ddot{\theta}}_i])^2 \; , \tag{4.36}$$

where $[\tilde{\ddot{\mathbf{d}}}_i, \tilde{\ddot{\theta}}_i]$ are the outputs predicted by the model.

Then, we start by creating a pool of 10^4 random combinations of $\ddot{G}_x$, $\ddot{G}_y$ and $\ddot{\phi}_z$. We divide this pool in four sets of 2500 input combinations, and we apply each set to a different configuration of the system:

1. The first configuration consists in a rectangular (2 × 3 [m]) mesh of mass–spring–damper elements (similar to the one in Fig. 4.1), with 77 nodes and stiffness, damping and nodal mass of 20 [N/m], 0.5 [N s/m] and 0.025 [kg] respectively. This object is held by four grippers at the corners of the rectangle.

2. The second configuration includes the same object than in the first one, but it is held by three grippers located as follows: two of them grasp the corners of one of the 3 [m] side, while the remaining one grasps the node in the middle of the opposing 3 [m] side of the mesh.

3. The third configuration includes the same object, but it is held by five grippers: four of them grasp the corners of the mesh, while the remaining one grasps the node in the middle of one of the 3 [m] side of the mesh.

4. The fourth configuration is the same than the first one except for the object: the stiffness of the mesh elements is set to 200 [N/m].

After that, the outputs $\ddot{d}_x$, $\ddot{d}_y$, $\ddot{d}_z$ and $\ddot{\theta}$ are obtained. The system configurations included in the pool of inputs and outputs provide sufficient depth for the analysis, since we consider similar cases in the next sections. The next step consists in creating an initial set of $S \geq 3$ inputs, by taking random samples from the pool. Algorithm 4.2 is then executed to maximize each observability index, by iteratively adding and removing inputs from the experiment design. Once the algorithm converges ($tolerance = 10^{-3}$ is chosen), the model is obtained from the selected inputs and the resulting outputs with least-squares.

Figure 4.2 shows the results of experiment designs in the range $3 \leq S \leq 50$. Due to the fact that the index optimization algorithm is affected by local maxima, the presented values are the mean of 20 maximization executions with different initial designs. Note that O_2, O_3,

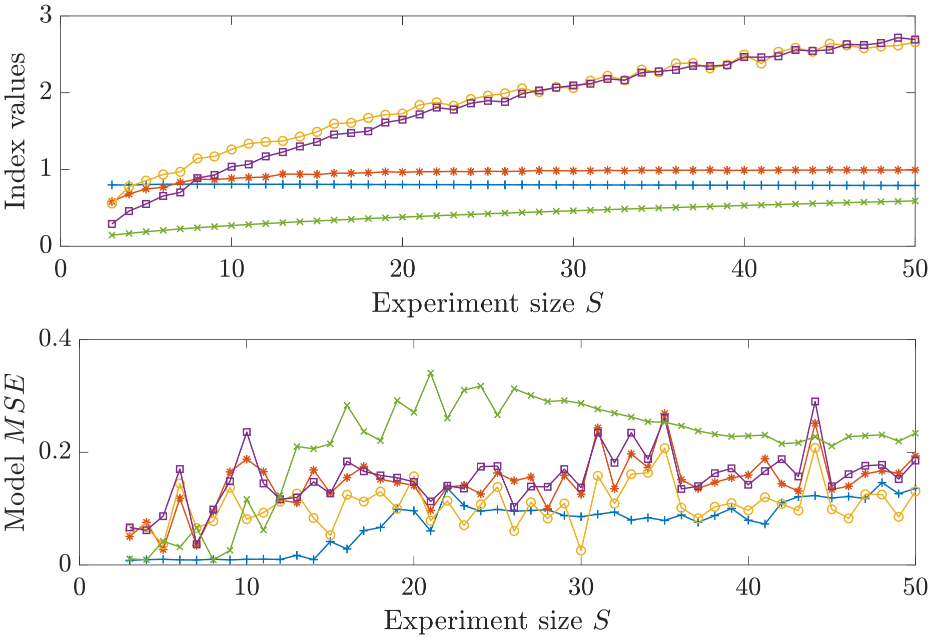


Figure 4.2: Comparison between the maximization of indexes O_1 (blue '+'), O_2 (orange '$*$'), O_3 (yellow '○'), O_4 (purple '□') and O_5 (green '×'), for experiment designs of increasing size S. In the upper plot, the mean observability index values of 20 simulations with different initial designs are depicted. In the lower plot, the mean values of the resulting models MSE, from the previous simulations, are shown.

O_4 and O_5 present monotonically increasing values, while O_1 first increases up to 0.81, for 7 measurements, and then shows a slight continuous decay, hardly noticeable due to the scale. The mean squared errors of the resulting models are also depicted. It can be seen that the errors obtained with the O_1 maximization are lower than the ones obtained with O_2, O_3 and O_4 in the interval $3 \leq S \leq 20$ (which is, in practice, our range of interest in the number of measurements), and very similar to the ones of O_5 at some points. In addition, the error of the model with O_1 follows the opposite direction to the evolution of the index values. This is the behavior we seek: the O_1 value evolution indicates when to stop taking measurements for getting a representative model with minimal error.

An additional test is carried out for verifying the effectiveness of O_1 and the maximization algorithm. Figure 4.3 represents the value of O_1 (thick dashed line) and the mean squared error of the resulting DBB model (thick blue line) after 20 index maximizations, which are executed with different initial experiment designs of size $S = 7$. For comparison, the values of O_1 (thin dotted line) and the mean squared error (thin dashed line) of 20 different random experiment designs with $S = 7$ are shown. The index-error pairs are sorted into increasing values of the error. We can see that the models obtained through index maximization are

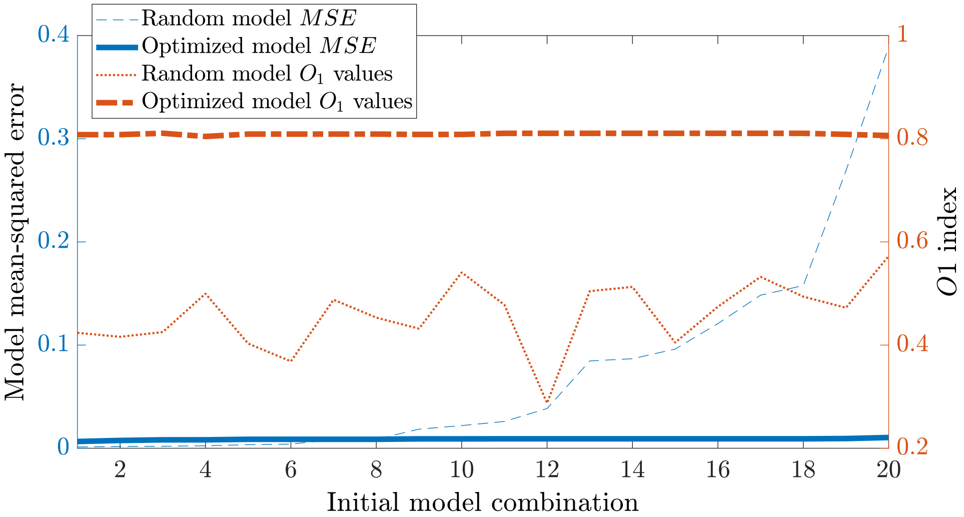


Figure 4.3: Comparison of 20 different experiments of size $S = 7$, created by maximizing the observability index O_1 (thick lines) and selecting random inputs (thin lines). The index values (orange) and the mean squared error of the models (blue) are shown, sorted into increasing values of the error.

much more homogeneous and more accurate. In addition, the system is explored in higher depth and, therefore, better characterized, as the observability index values indicate.

4.5.2 Test with static obstacles

We evaluate our proposal in two different simulation scenarios in Matlab®. The first scenario includes three static obstacles, and allows us to compare the QP controller with our previous method for agents with single integrator dynamics [62]. We deploy a team of $N = 4$ robots grasping a 2×3 [m] rectangular sheet by its four corners, at 1.7 [m] height. The sheet is modeled with a mesh of mass-spring-damper elements of 20 [N/m] stiffness, 0.5 [N·s/m] damping and 77 nodes with a 0.025 [kg] nodal mass. Note that these properties are not known by the system. Instead, we obtain the DBB model by maximizing the O_1 index from a set of $S = 7$ measurements. The control parameters are set as follows: $k_{1d} = 0.4$, $k_{2d} = 1$, $k_{1r} = 1$, $k_{2r} = 2.5$, $k_{1t} = 0.01$, $k_{2t} = 0.3$, $\varepsilon = 0.5\,h_{lj}$, $\alpha_l = 10$ [m/s^2], $D_{l1}^{min} = 0.55$ [m], $D_{l2}^{min} = 0.95$ [m], $D_{l3}^{min} = 0.65$ [m], $k_{w1} = k_{w2} = 0.2$, $k_{w3} = 0.1$, $k_{w4} = k_{w5} = 0.25$ and $R = 4$ [m]. The limit values to avoid overstretching and excessive velocities are set as $\mathbf{d}^{max} = [4, 6, 1.5]$ [m], $\mathbf{d}^{min} = [0.5, 1, 0.1]$ [m], $\dot{\theta}^{max} = -\dot{\theta}^{min} = \pi$ [rad/s] and $\dot{c}_x^{max} = \dot{c}_y^{max} = -\dot{c}_x^{min} = -\dot{c}_y^{min} = 0.5$ [m/s]. It is worth noting that fine tuning of these parameters is not needed to successfully complete the task. In addition, we define $L = 488$ evenly distributed virtual points over the BB faces. The goal of the task is to transport the sheet without collisions to a target position in [12, 5] meters, while getting the initial dimensions and

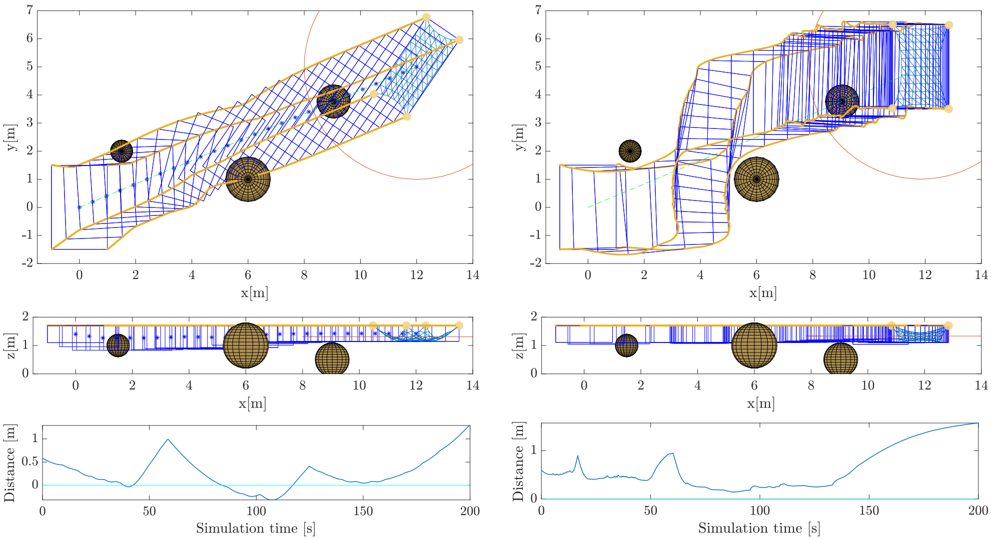


Figure 4.4: The goal is to steer, without collisions, the object (blue mesh) from the initial position at [0, 0] meters to the target position at [12, 5] meters. Images at the left side correspond to our previous method [62], while at the right side we show the new method with the QP controller. The top and middle images show the top and side views of the test circuit, with the grippers at the corners of the sheet and their trajectories in orange. The bounding boxes of the object over time (blue boxes), the obstacles (brown spheres), the sensors range (orange circumference) and the ideal transport route (green dashed line) are also depicted. The bottom images show the absolute minimum distances between the mesh nodes and the nearest obstacle center. We subtract the radius of the nearest obstacle to check whether the node is inside (negative distance) or outside the obstacle (positive distance). We can see that our previous method is unable to avoid collisions at some instants, and we have no control over the shape and the orientation of the object. In contrast, the proposed method prevents collisions with the obstacles and preserves the initial dimensions and orientation of the box.

orientation of the BB at the end of the test. Figure 5.2 shows the performance of our method in the referred task compared with the previous work. We can see that the QP controller is able to achieve the desired configuration of the object without collisions, while our previous approach fails in the latter purpose and it does not allow controlling the shape and orientation of the object. The bottom plots, which display the minimum distance between any mesh node and the nearest detected obstacle, illustrate the failure of the previous method and the success of the new one. When the value is negative, it means the node has penetrated that distance inside the volume of the obstacle. From this test we see that the impact of the imposed route in our previous method is drastically reduced in the current proposal, with a much more flexible and robust formulation.

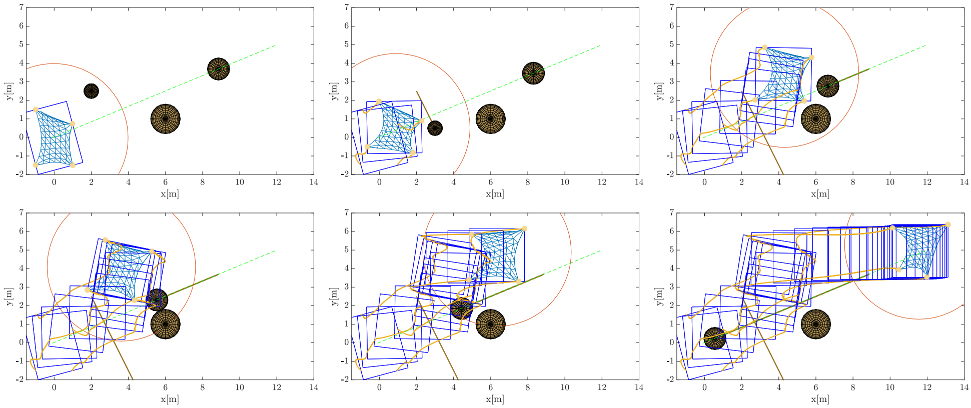


Figure 4.5: From top to bottom and left to right, we show six instants (t = 0,10,40,60,80,150 [s]) of the second test. The system reaches the position [12, 5] meters while avoiding collision with two dynamic and one static obstacles (see the video for full sequence).

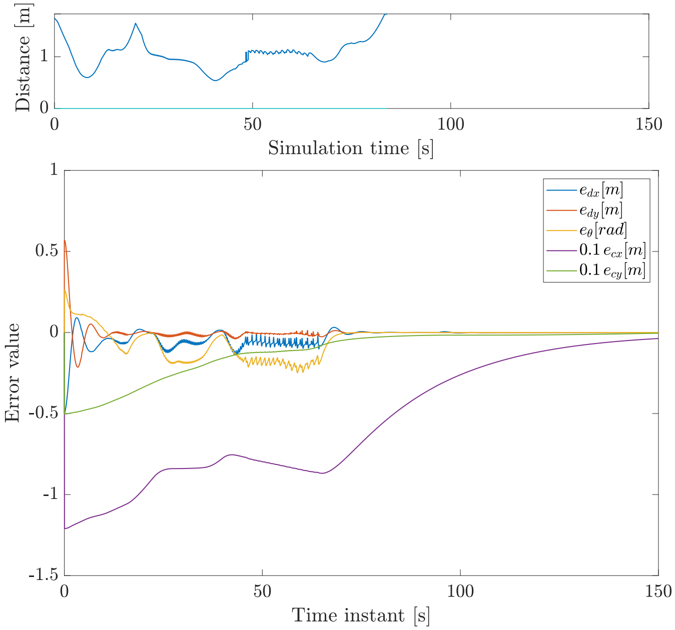


Figure 4.6: At the top, absolute minimum distances between the mesh nodes and the nearest obstacle, at different time instants in the second test case. At the bottom, values of the deformation, orientation and position control errors. Note that the position error $\mathbf{e}_d$ is scaled by 0.1.

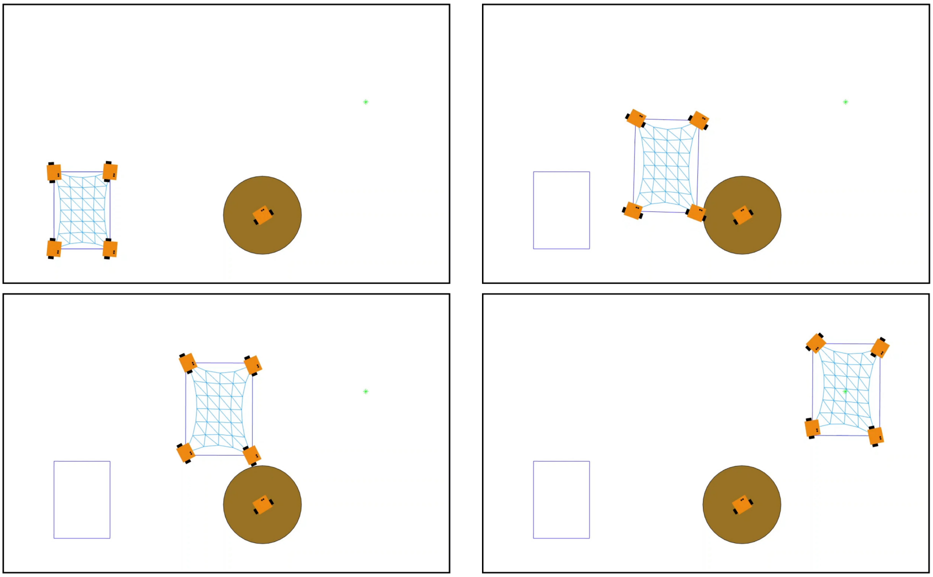


Figure 4.7: From top to bottom and left to right, we show four instants (t = 0,100,150,450 [s]) of the simulation with realistic conditions. We can see that the virtual object is driven to the desired configuration while avoiding collisions with the static robot, placed between the initial and the final position.

4.5.3 Test with dynamic obstacles

In the second test scenario we evaluate the system's performance under dynamic obstacles. Again, a team of $N = 4$ robots grasps a deformable sheet with the same mechanical properties than in the previous test, but in a different configuration. The DBB model is obtained following the same procedure, and the control parameters are equal except $k_{1d} = 1.0$, $k_{2d} = 2.0$, $k_{1r} = 1.5$, $k_{2r} = 3.0$, $\alpha_l = 15$ [m/s^2]. With these changes we get a faster behavior of the nominal controller, and the system reacts better to the incoming obstacles. The goal is to transport the sheet without collisions to the target position in [12, 5] meters, while simultaneously reducing a 20% the vertical y-dimension of the BB, increasing a 20% the x-dimension and getting a 0 degrees orientation of the BB. In this case, we consider two dynamic and one static obstacles. Figure 5.3 shows six different snapshots, while Fig. 4.6 shows the absolute minimum distances between the mesh nodes and the nearest obstacle as well as the control errors. Despite the small bounces starting at 50 [s] and stopping around 70 [s], when the second obstacle approaches the system, we can see that the object reaches the desired configuration without collisions. Note that the robots stop detecting obstacles at t = 84 [s], which is the reason why the minimum distances plot shows no values from there on. The average computational time of the algorithm is 0.016 [s], which seems fast enough for a wide range of practical cases with real time response. This fact is verified with the results we show in the next section.

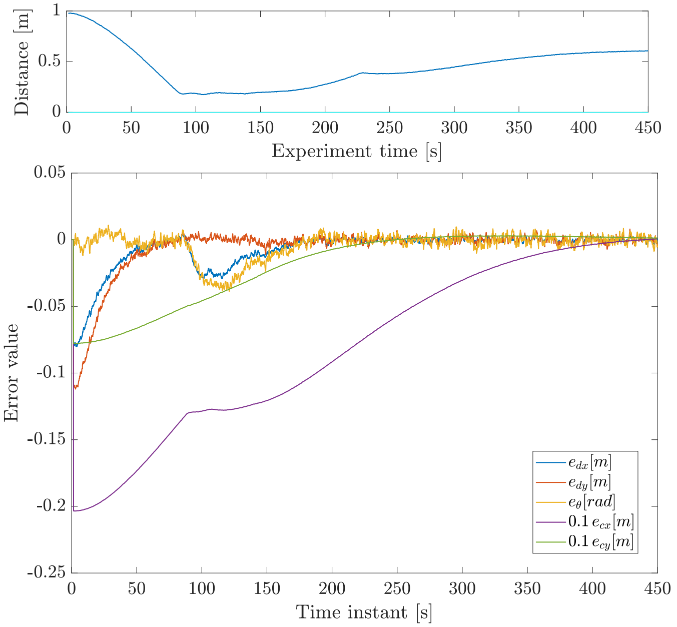


Figure 4.8: At the top, absolute minimum distances between the nodes representing the object and the obstacle, at different time instants in the realistic case. At the bottom, components of the deformation, orientation and position errors. The position error $\mathbf{e}_d$ is again scaled by a 0.1 factor.

4.5.4 Test with realistic conditions

Next we tackle a transport task in a scenario with realistic conditions. The goal of the task consists in transporting a 0.40×0.55 [m] rectangular cloth-like object by its four corners to a specific position, preserving the initial orientation of the BB and expanding a 20% its x-y dimensions. We add random noise in the position of the robots, with 0.01 [m] amplitude, and the robots' linear velocities are saturated to 0.05 [m/s]. The algorithm also runs with real-time measurements. It is worth to mention that the robots follow unicycle kinematics in this case, instead of double-integrator ones. Therefore, we integrate the acceleration outputs and then transform the resulting velocities to the unicycle model by means of a diffeomorphism [134]. An agent acting as a static obstacle is placed between the initial and the goal positions. The control parameters are set in the following manner: $k_{1d} = 0.02$, $k_{2d} = 0.5$, $k_{1r} = 0.01$, $k_{2r} = 0.3$, $k_{1t} = 0.0005$, $k_{2t} = 0.06$, $\varepsilon = 5000\, h_{lj}^3$, $\alpha_l = 0.05$ [m/s^2], $D_{l1}^{min} = 0.14$ [m], $k_{w1} = k_{w2} = k_{w3} = k_{w4} = k_{w5} = 0.2$ and $R = 5$ [m]. The limit values to avoid overstretching and excessive velocities are set as $\mathbf{d}^{max} = [0.6, 0.9, 1]$ [m], $\mathbf{d}^{min} = [0.15, 0.15, 0.01]$ [m], $\dot{\theta}^{max} = -\dot{\theta}^{min} = 1.8$ [rad/s] and $\dot{c}_x^{max} = \dot{c}_y^{max} = -\dot{c}_x^{min} = -\dot{c}_y^{min} = 0.05$ [m/s]. In addition, we define $L = 152$ evenly distributed virtual points over the BB faces.

Figure 4.7 shows four different time instants of the simulation. We can see that the robots are able to drive the virtual deformable object from the initial position, at the bottom left side, to the final position at the top right side of the arena, avoiding the brown area (with 0.14 [m] radius) around the obstacle robot in the process. Figure 4.8 shows the minimum node-obstacles distances, always above cero, and the control errors, which tend to zero over time. These results confirm that the proposed DBB model is valid and useful for computing the dimensions, orientation and position of the BB under the control actions. Also the QP controller we present exploits the model successfully to achieve the control goal in all the tested cases.

4.6 Discussion

We have presented a method for achieving a desired configuration of a large deformable object in terms of position, dimensions and orientation of its bounding box, without collisions. Our approach is based on a 3D deformable bounding box model for mobile robots that manipulate the object with double-integrator dynamics. It allows predicting the object evolution in shape and orientation under specific control inputs. We exploit this model to formulate a set of nominal controllers, which steer the object to the target configuration. Then, we propose a control barrier function that guarantees collision avoidance by means of an integrated quadratic-programming controller. This controller includes additional linear constraints that limit the control inputs, so that the bounding box dimensions and the translation and rotation velocities remain in the admissible ranges. Simulation results show that our method successfully completes different kinds of transport tasks, in which the objects are steered to specific configurations in environments with static and dynamic obstacles.

Modelling deformation is challenging. The control strategy may be limited if it relies on a model that is not accurate, and the manipulators could show erratic behaviors. In the next chapter, we apply formation control to the multirobot system that handles the deformable object. By doing so, we can define the goal configuration based on the properties of the robotic formation (shape, scale, position and orientation), thus bypassing the issues related to modelling deformation. Besides, we show that the resulting controller offers strong guarantees in terms of uncoupling of the formation parameters, stability and convergence.

Chapter 5

Formation control for transport tasks of deformable objects

We present in this chapter a formation controller that allows manipulating deformable objects, by means of a team of mobile robots modelled with double-integrator dynamics. The manipulation task is defined as reaching a target configuration consisting of a desired shape, scale, position and orientation of the formation in 2D, while preserving the integrity of the object. We provide a set of controllers, whose formal analysis is covered in depth, designed to allow the uncoupled control of the variables that define the task. Besides, we include control barrier functions to enforce collision and excessive stretching avoidance. The performance of the method is illustrated in simulation and in experiments with real robots.

5.1 Introduction

SUBJECT to manipulation, real objects show a wide variety of mechanical behaviors, particularly in terms of deformation. This means that finding general deformation models is, indeed, a challenging task. In addition, as previously stated, multirobot systems for manipulation of deformable objects require accurate and tightly coordinated motions of the involved robotic platforms. Our proposal for solving both issues is a formation controller that avoids deformation modelling.

Formation control methods seem an appropriate basis for consideration, since they put the focus in controlling the global behavior of a team of robots, for achieving a common goal. Being one of the most actively studied topics in the area of multirobot systems, formation control covers a wide variety of approaches [106]. These methods can be classified depending on the dynamic model of the robots they consider. Aranda *et al.* exploited a formation controller for robots with single-integrator dynamics, for performing manipulation tasks of deformable objects [11]. In this work, the control goal is defined as the combination of shape, orientation, centroid and distances between robots. The distributed pipeline by Lusk *et al.* also considers the single-integrator model, for flying units in collision-free formations [93].

Other studies develop formation controllers for robots with higher-order models. Necessary and sufficient conditions for stability of an affine formation controller are provided by Lin *et al.* [87]. This method, formulated for robots with single-integrator dynamics and extended to robots with second-order dynamics, allows reaching a configuration where collinearity and ratios of distances with respect to a reference configuration are preserved. A different study by Fathian *et al.* proposes a distributed control strategy where the robots are modeled with single-integrator and higher-order holonomic dynamics [46]. In their work, communication between the robots or a common reference for orientation are not needed. This idea is further developed for robots with higher-order and non-holonomic dynamics, with a distributed barycentric-coordinate-based controller for the purpose of collision-free formation control. Dimarogonas and Kyriakopoulos proposed a formation controller for robots with single-integrator and unicycle models and analyzed the connection between formation infeasibility and velocity alignment [41]. Steering the robots to a desired geometric pattern and achieving desired collective maneuvers is the goal of the work proposed by Zhao [139]. The control laws he presents, based on stress matrices and with leader-follower architecture, are formulated for robots with single-integrator, double-integrator and unicycle models. Another study by Zhao *et al.* shows a general approach for coordinating multiple robots with motion constraints, in two and three dimensions by means of a gradient-descent control law [140]. A transportation strategy for robots with double-intergrator dynamics, with graph-based path planning and model predictive control (MPC), is showed by Ebel *et al.* [44]. Their method is compatible with variable numbers of robots in the formation, and includes a memory functionality for finding exit paths in maze-like environments.

Different works have accomplished tasks where deformable objects are manipulated. A switching control strategy for differential drive mobile robots is proposed by Felix-Rendon *et al.* for the purpose of controlling the shape of a deformable object [47]. Shape servoing is a well-known alternative for manipulation of deformable objects. This area of study includes a method based on the As-Rigid-As-Possible (ARAP) model [115], where previous information about the mechanical properties of the object is not needed. Planning strategies have also been developed for manipulating deformable objects with multiple manipulators. In this field, we can find hybrid centralized-distributed approaches with high-level global guidance and receding horizon local planners, for collision avoidance and shape maintenance [5]. Also the problem of towing a rigid load with cables attached to multiple quadrupedal robots has been solved with a centralized-distributed planner [135]. Alternating planning and control by means of a deadlock prediction system is another proposal to solve the problem of manipulating deformable objects [95]. The problem of transporting a rigid object with a deformable sheet, held by multiple mobile manipulators, has received attention in the last years [67], [71]. Gripper positioning in shape control tasks of deformable objects is an interesting problem that has been recently tackled, through optimization of an importance metric [32].

Finally, different alternative systems for transporting deformable objects to goal configurations have been previously proposed. A system for simultaneous shape control and transport of a 2D deformable object is proposed by Lopez-Nicolas *et al.* for robots with non-holonomic motion constraints [91]. In this work, smooth admissible trajectories are followed so that the integrity of the object is preserved, but collision avoidance is not tackled. The object deformation is obtained from a Deformable Bounding Box model (DBB) in another method for transporting deformable objects to a goal configuration [64]. In contrast to the present strategy, the configuration parameters are limited to the dimensions, rotation and position of the DBB, and the stability and convergence of the method are not formally analyzed.

We build this chapter upon a previous study [57]. In the previous work, we proposed a novel formation controller for robots with double integrator dynamics, with a set of CBFs that allow avoiding overstretching, robot-to-robot, robot-to-obstacle and object-to-obstacle collisions. This work was based, in turn, in a formation controller for robots with single-integrator kinematics by Aranda *et al.* [11]. The extension of this work to robots with double-integrator dynamics was interesting for different reasons. Despite the technical challenges of this extension (more complex structure of the controllers, more difficult analysis of the control properties and integration of the system), it allowed including inertial effects to the system, which are important in many practical scenarios. It also brought about the potential to control the robot-object contact forces. Besides, we augmented the system with the CBFs for collision avoidance, where controlling accelerations is a key aspect [132], and overstretching avoidance. The main contributions we present here, with respect to the previous studies, are a new control term that guarantees full uncoupling between the controlled formation parameters and a comprehensive formal analysis of uncoupling, stability and convergence of the proposal. New simulations and experimental results allow us to illustrate the performance of our method.

5.2 Problem statement

Let us consider a deformable object that is carried by a formation of N robots in $\mathbb{R}^2$. The object is connected to the robots by a set of points $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_N] \in \mathbb{R}^{2\times N}$, where $\mathbf{p}_i = [p_{ix}, p_{iy}]^\intercal$, $i = 1, ..., N$ denotes the center of robot i. We model the robot-object links as free rotating joints. The formation centroid is $\mathbf{g} = \frac{1}{N}\mathbf{P}\,\mathbf{1}_N$, with $\mathbf{1}_N$ being a column vector of N ones. s and θ denote the scale and orientation of the formation, respectively. We consider the double-integrator model to describe the robots' dynamics:

$$\begin{bmatrix} \dot{\mathbf{p}}_i \\ \ddot{\mathbf{p}}_i \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{2\times 2} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{p}_i \\ \dot{\mathbf{p}}_i \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_{2\times 2} \end{bmatrix} \mathbf{u}_i \;, \tag{5.1}$$

where $\mathbf{x}_i = [\mathbf{p}_i^\intercal, \dot{\mathbf{p}}_i^\intercal]^\intercal$ is the state of robot i and $\mathbf{u}_i = [u_{ix}, u_{iy}]^\intercal$, $i = 1, ..., N$ is its control input.

The goal task consists in driving the deformable object to a target configuration, understood as a specific combination of shape, scale, position and orientation of the robotic formation. We adopt a strategy to solve this problem such that the configuration of the object is not controlled explicitly. Instead, the controllers we propose act over the formation of robots that manipulate the object. Our method is especially suited for highly deformable objects, whose structure is able to follow closely the shape of the formation. The goal configuration of the formation is encoded by a set of variables. We define the desired shape of the formation as the matrix $\mathbf{P}_d = [\mathbf{p}_{d1}, \mathbf{p}_{d2}, ..., \mathbf{p}_{dN}] \in \mathbb{R}^{2\times N}$, where $\mathbf{p}_{di} = [p_{dix}, p_{diy}]^\intercal$, $i = 1, ..., N$ are the positions of the robots in the desired shape. Using the desired formation scale s_d and the desired formation orientation θ_d, we can express the target configuration as

$$\mathbf{P}_T = s_d \mathbf{R}_d(\theta_d)(\mathbf{P}_d - \mathbf{g}_d \mathbf{1}_N^\intercal) + \mathbf{g}_d \mathbf{1}_N^\intercal \ , \tag{5.2}$$

where $\mathbf{R}_d(\theta_d) \in SO(2)$ is the 2D rotation matrix corresponding to the angle θ_d. Without loss of generality we consider that the desired shape is centered around the desired formation centroid $\mathbf{g}_d = \frac{1}{N}\mathbf{P}_d \mathbf{1}_N$, and $\theta_d = 0$.

Due to the fact that we consider an environment with obstacles, reaching the target configuration requires robot-to-obstacle and object-to-obstacle collision avoidance.

5.3 Formation control for robots with double-integrator dynamics

For controlling the variables of interest in our problem, we present a set of controllers for robots modelled with double-integrator dynamics. The control design is such that every control term is focused on a specific variable, trying not to affect the rest.

5.3.1 Shape control

First, we will describe the controller we propose for achieving the desired shape of the formation. This controller is based on a cost function that defines the shape error of the robotic formation relative to the desired shape [11]:

$$\gamma = \frac{1}{2}||\mathbf{P}_b - \mathbf{H}\mathbf{P}_{db}||_F^2 = \frac{1}{2}\mathrm{tr}\Big((\mathbf{P}_b - \mathbf{H}\mathbf{P}_{db})^\intercal(\mathbf{P}_b - \mathbf{H}\mathbf{P}_{db})\Big) \ . \tag{5.3}$$

where $||\cdot||_F$ denotes the Frobenius norm and tr denotes the trace. The matrix $\mathbf{H} \in \mathbb{R}^{2\times 2}$ in (5.3), assumed $\neq \mathbf{0}$, is defined as $\mathbf{H} = [(h_1, h_2)^\intercal, (-h_2, h_1)^\intercal]$. where

$$h_1 = \frac{\mathrm{tr}(\mathbf{P}_b\mathbf{P}_{db}^\intercal)}{\mathrm{tr}(\mathbf{P}_{db}\mathbf{P}_{db}^\intercal)} \ , \quad h_2 = \frac{\mathrm{tr}(\mathbf{P}_b(\mathbf{S}\mathbf{P}_{db})^\intercal)}{\mathrm{tr}(\mathbf{P}_{db}\mathbf{P}_{db}^\intercal)} \ , \tag{5.4}$$

where $\mathbf{S} = [(0, 1)^\intercal, (-1, 0)^\intercal]$. Note that $\mathbf{S}$ is a 2D rotation matrix of $\pi/2$ radians. The matrix $\mathbf{H}$ aligns the matrices $\mathbf{P}_b$ and $\mathbf{P}_{db}$ by performing rotation and uniform scaling. $\mathbf{P}_b$ and $\mathbf{P}_{db}$ represent, respectively, the current and desired shape of the formation with zero centroid, and are defined as

$$\mathbf{P}_b = \mathbf{P} - \mathbf{g}\,\mathbf{1}_N^\intercal = \mathbf{P}\,\mathbf{K}_b \ , \tag{5.5}$$

$$\mathbf{P}_{db} = \mathbf{P}_d - \mathbf{g}_d \mathbf{1}_N^\intercal = \mathbf{P}_d\,\mathbf{K}_b \ , \tag{5.6}$$

with $\mathbf{K}_b = \mathbf{I}_{N\times N} - (1/N)\,\mathbf{1}_N\mathbf{1}_N^\intercal$. Observe that $\mathbf{K}_b$ is a centering matrix that satisfies $\mathbf{K}_b^\intercal = \mathbf{K}_b$, $\mathbf{K}_b\mathbf{1}_N = \mathbf{0}$ and $\mathbf{K}_b\mathbf{K}_b = \mathbf{K}_b$. We also define $\mathbf{E}_\gamma \in \mathbb{R}^{2\times N}$ as the matrix of the position errors relative to the desired shape:

$$\mathbf{E}_\gamma = \mathbf{P}_b - \mathbf{H}\mathbf{P}_{db}. \tag{5.7}$$

Therefore, we can also express γ as follows:

$$\gamma = \frac{1}{2}\mathrm{tr}(\mathbf{E}_\gamma^\intercal\,\mathbf{E}_\gamma). \tag{5.8}$$

Then, we build the controller from (5.3) as the linear combination of the negative gradient of γ and the time derivative of the negative gradient of γ:

$$\begin{aligned}\mathbf{U}_H &= -k_{1H}\nabla_{\mathbf{P}}\gamma - k_{2H}\frac{d(\nabla_{\mathbf{P}}\gamma)}{dt} = \\ &= k_{1H}(\mathbf{H}\mathbf{P}_{db} - \mathbf{P}_b) + k_{2H}(\dot{\mathbf{H}}\mathbf{P}_{db} - \dot{\mathbf{P}}_b)\ ,\end{aligned} \tag{5.9}$$

being k_{1H} and k_{2H} positive control gains. Inspired by the control scheme proposed by Fathian *et al.* [46, eq. (22)], this controller aims at optimally driving the formation so that the cost function and its time derivative are reduced. This implies a direct reduction of the difference in shape between $\mathbf{P}_b$ and $\mathbf{P}_{db}$. Due to the fact that there is no direct control over the transition from $\mathbf{P}_b$ and $\mathbf{P}_{db}$, if the difference between them is large, $\mathbf{U}_H$ may temporarily reorder the robots around the object. For example, if the formation consists of three robots following a clockwise ordering $\{1, 2, 3\}$, $\mathbf{U}_H$ may modify the sequence to $\{2, 1, 3\}$ or $\{1, 3, 2\}$. This reordering could fold or twist the object that is being manipulated by the formation, causing undesired deformations or structural damage. For avoiding these undesired effects, we formulate a correcting term

$$\mathbf{U}_G = k_{1G}(\mathbf{P}_b\mathbf{P}_{db}^{+}\mathbf{P}_{db} - \mathbf{P}_b) + k_{2G}(\dot{\mathbf{P}}_b\mathbf{P}_{db}^{+}\mathbf{P}_{db} - \dot{\mathbf{P}}_b)\ , \tag{5.10}$$

where

$$\mathbf{P}_{db}^{+} = \mathbf{P}_{db}^\intercal(\mathbf{P}_{db}\mathbf{P}_{db}^\intercal)^{-1} \tag{5.11}$$

represents the Moore-Penrose pseudoinverse of matrix $\mathbf{P}_{db}$ and k_{1G} and k_{2G} are positive control gains. This term is based on an optimal affine transformation that aligns $\mathbf{P}_{db}$ with $\mathbf{P}_b$ in a least-squares manner. $\mathbf{U}_G$ steers the system towards $\mathbf{P}_b\mathbf{P}_{db}^{+}\mathbf{P}_{db}$, which is the optimal affine deformation of $\mathbf{P}_{db}$. We denote by $\mathbf{G} \in \mathbb{R}^{2\times 2}$ the matrix that expresses this deformation:

$$\mathbf{G} = \mathbf{P}_b\mathbf{P}_{db}^{+}. \tag{5.12}$$

Therefore, $\mathbf{U}_G$ can also be expressed as

$$\mathbf{U}_G = k_{1G}(\mathbf{G}\mathbf{P}_{db} - \mathbf{P}_b) + k_{2G}(\dot{\mathbf{G}}\mathbf{P}_{db} - \dot{\mathbf{P}}_b)\ . \tag{5.13}$$

For numerical reasons, $\mathbf{U}_G$ is compatible with desired shapes different from a straight line, and it limits the movements of the robots to those that produce the deformation modes of stretch and shear [11]. Note that it is still possible to manipulate rope-like objects in straight-line configurations with the method we propose, and this control term would be unnecessary. We define another variable to use in our analysis: $\mathbf{E}_{GH} \in \mathbb{R}^{2\times N}$, which measures the difference between the configurations that result from applying the two transformation matrices $\mathbf{G}$ and $\mathbf{H}$:

$$\mathbf{E}_{GH} = \mathbf{G}\mathbf{P}_{db} - \mathbf{H}\mathbf{P}_{db}. \tag{5.14}$$

We combine the previous controllers in

$$\mathbf{U}_{\gamma} = \alpha_H \mathbf{U}_H + \alpha_G \mathbf{U}_G \ . \tag{5.15}$$

The positive control weights α_H and α_G regulate the contribution of each term, so that α_G should be greater than α_H if there is a large difference between $\mathbf{P}_b$ and $\mathbf{P}_{db}$. Otherwise, the convergence speed can be increased with a greater value of α_H. As shown in Section 5.5, this shape controller steers γ to zero. This makes the team acquire the shape of $\mathbf{P}_d$.

5.3.2 Scale control

Scaling the shape of the formation consists in uniformly driving the robots closer or further to the formation centroid. Just like $\mathbf{U}_{\gamma}$, this process creates deformations over the object, since it modifies the relative positions of the robots that grasp it. The formation scale can be obtained as $s = ||\mathbf{H}||_2$. For getting $s = s_d$, we propose the controller

$$\mathbf{U}_s = -k_{1s} e_s (1/s) \mathbf{H}\mathbf{P}_{db} - k_{2s} \dot{\mathbf{P}}_b \ , \tag{5.16}$$

where $e_s = s - s_d$ represents the scale error and k_{1s} and k_{2s} are positive control gains. This controller produces a uniform scaling of the goal shape $\mathbf{H}\mathbf{P}_{db}$, which is proportional to the scale error and the velocity of the robots.

Assumption 5.1. *We assume scale $s > 0$ initially (t=0). As we will show later, we can control the dynamics of s independently from the rest of the variables with $\mathbf{U}_s$. Therefore, by selecting the control gains appropriately, we can guarantee $s > 0, \forall t \geq 0$, with a monotonic convergence to s_d. Note that this requirement is necessary for the orientation of the formation θ to be defined at all times.*

5.3.3 Translation and rotation control

Translation and rotation of the robotic formation consist in driving the formation rigidly from one place to another and rotating the shape around the formation centroid, respectively. We can obtain the orientation of the formation as $\theta = \text{atan2}(h_2, h_1) \in (-\pi, \pi]$, where the atan2 operator represents the four quadrant inverse tangent. Note that θ can be assumed to always remain differentiable in terms of h_1 and h_2, as done in [11]. The controllers we apply to get these transformations are

$$\mathbf{U}_g = -k_{1g}\mathbf{e}_g\mathbf{1}_N^\intercal - k_{2g}\dot{\mathbf{P}} \ , \tag{5.17}$$

$$\mathbf{U}_\theta = -k_{1\theta}e_\theta\mathbf{SHP}_{db} - k_{2\theta}\dot{\mathbf{P}}_b \ , \tag{5.18}$$

with $\mathbf{e}_g = \mathbf{g} - \mathbf{g}_d$ being the translation error, $e_\theta = \theta - \theta_d$ being the orientation error, and k_{1g}, k_{2g}, $k_{1\theta}$ and $k_{2\theta}$ being positive control gains.

5.3.4 Full formation controller

The full formation controller consists in a linear combination of the shape, scale, translation and rotation controllers for robots with double-integrator dynamics ($\mathbf{U}_\gamma + \mathbf{U}_s + \mathbf{U}_g + \mathbf{U}_\theta$). This control law provides suitable performance. However, when using it, the dynamic evolution of the variables s and θ is coupled. The underlying reason for this coupling can be found by computing the dynamics of these variables as a function of h_1 and h_2. Using standard manipulations and also the facts that $h_1 = s\cos\theta$ and $h_2 = s\sin\theta$, we get:

$$\dot{s} = \frac{h_1\dot{h}_1 + h_2\dot{h}_2}{s}, \quad \ddot{s} = \frac{h_1\ddot{h}_1 + h_2\ddot{h}_2}{s} + \dot{\theta}^2 s. \tag{5.19}$$

$$\dot{\theta} = \frac{h_1\dot{h}_2 - h_2\dot{h}_1}{s^2}, \quad \ddot{\theta} = \frac{h_1\ddot{h}_2 - h_2\ddot{h}_1}{s^2} - \frac{2\dot{s}\dot{\theta}}{s}. \tag{5.20}$$

The second addends in the second-order time derivatives of s and of θ cause the coupling. Specifically, it turns out that s is not invariant when using $\mathbf{U}_\theta$, as we analyze in Section 5.4. To remove this coupling, we define a control term $\mathbf{U}_u$ having the following form:

$$\mathbf{U}_u = -\dot{\theta}^2\mathbf{HP}_{db} + (2\dot{s}\dot{\theta}/s)\mathbf{SHP}_{db} \ . \tag{5.21}$$

The first addend of $\mathbf{U}_u$ aims at modifying the dynamics of $\mathbf{U}_s$, while the second one does that for $\mathbf{U}_\theta$. The full control law we apply is, then:

$$\mathbf{U} = \mathbf{U}_\gamma + \mathbf{U}_g + \mathbf{U}_s + \mathbf{U}_\theta + \mathbf{U}_u \ . \tag{5.22}$$

As we show in Section 5.5, $\mathbf{U}_u$ will cancel out the coupled dynamic terms. Then, under the control (5.22), we will obtain uncoupled dynamics for all the variables (γ, θ, $\mathbf{g}$ and s).

5.4 Invariance and uncoupling under the proposed control terms

Uncoupling is a desired property for the controller we propose. When it applies, it is possible to control each variable independently. This results in flexible solutions, which can be adapted to different manipulation tasks. It is also possible to study other control properties of every term separately (e.g., stability), if uncoupling between them exists. To study uncoupling, we analyze in this section the invariance of each variable (γ, $\mathbf{g}$, θ and s) under the proposed controllers. Note that we analyze the role of $\mathbf{U}_u$ to fully uncouple the control of the formation variables in Section 5.5. In our analysis we will assume the system is at rest initially, i.e., $\dot{\mathbf{P}}(t=0) = \mathbf{0}$. At the end of this section (Remark 5.1), we will summarize the results of our invariance study.

Proposition 5.1. *The shape variable γ is invariant under $\mathbf{U}_g$, $\mathbf{U}_s$ and $\mathbf{U}_\theta$.*

Proof. We will use the time derivative of γ, which has the following expression:

$$\dot{\gamma} = \operatorname{tr}\Big(\mathbf{P}_b^\intercal \dot{\mathbf{P}}_b - \mathbf{P}_{db}^\intercal \mathbf{H}^\intercal \dot{\mathbf{P}}_b - \mathbf{P}_{db}^\intercal \dot{\mathbf{H}}^\intercal \mathbf{P}_b + \mathbf{P}_{db}^\intercal \dot{\mathbf{H}}^\intercal \mathbf{H} \mathbf{P}_{db}\Big) \,. \tag{5.23}$$

We will now find a simpler expression for this derivative. First, we know from Aranda *et al.* [11] that

$$\mathbf{H}\mathbf{P}_{db} = h_1 \mathbf{P}_{db} + h_2 \mathbf{S}\mathbf{P}_{db} \,, \tag{5.24}$$

and hence

$$\dot{\mathbf{H}}\mathbf{P}_{db} = \dot{h}_1 \mathbf{P}_{db} + \dot{h}_2 \mathbf{S}\mathbf{P}_{db} \,. \tag{5.25}$$

Using these two identities, the definitions (5.4), and applying the property $\operatorname{tr}((\mathbf{S}\mathbf{A})\mathbf{A}^\intercal) = \operatorname{tr}((\mathbf{S}\mathbf{A})^\intercal \mathbf{A}) = 0$ with $\mathbf{A} \in \mathbb{R}^{2\times N}$ for $\mathbf{A} = \mathbf{P}_{db}$, we find the identity

$$\begin{aligned}
&\operatorname{tr}(\mathbf{P}_{db}^\intercal \dot{\mathbf{H}}^\intercal \mathbf{H} \mathbf{P}_{db}) = \\
&= \operatorname{tr}\Big((\dot{h}_1 \mathbf{P}_{db} + \dot{h}_2 \mathbf{S}\mathbf{P}_{db})^\intercal (h_1 \mathbf{P}_{db} + h_2 \mathbf{S}\mathbf{P}_{db})\Big) \\
&= h_1 \dot{h}_1 \operatorname{tr}(\mathbf{P}_{db}\mathbf{P}_{db}^\intercal) + h_2 \dot{h}_2 \operatorname{tr}(\mathbf{P}_{db}^\intercal \mathbf{S}^\intercal \mathbf{S} \mathbf{P}_{db}) \\
&= h_1 \operatorname{tr}(\dot{\mathbf{P}}_b^\intercal \mathbf{P}_{db}) + h_2 \operatorname{tr}(\dot{\mathbf{P}}_b^\intercal \mathbf{S}\mathbf{P}_{db}) = \operatorname{tr}(\mathbf{P}_{db}^\intercal \mathbf{H}^\intercal \dot{\mathbf{P}}_b) \,.
\end{aligned} \tag{5.26}$$

Using this on (5.23), we reach the simpler expression

$$\dot{\gamma} = \operatorname{tr}(\mathbf{P}_b^\intercal \dot{\mathbf{P}}_b - \mathbf{P}_{db}^\intercal \dot{\mathbf{H}}^\intercal \mathbf{P}_b) \,. \tag{5.27}$$

We present the study for each controller next.

1) γ is invariant under $\mathbf{U}_g$.

With $\ddot{\mathbf{P}} = \mathbf{U}_g$, we have $\ddot{\mathbf{P}}_b = \ddot{\mathbf{P}}\mathbf{K}_b = \mathbf{U}_g \mathbf{K}_b = -k_{1g}\mathbf{e}_g \mathbf{1}_N^\intercal \mathbf{K}_b - k_{2g}\dot{\mathbf{P}}_b$. Since $\mathbf{1}_N^\intercal \mathbf{K}_b = (\mathbf{K}_b \mathbf{1}_N)^\intercal = \mathbf{0}$ and assuming initial rest (i.e., $\dot{\mathbf{P}}(t=0) = \mathbf{0}$), we see that $\ddot{\mathbf{P}}_b = \dot{\mathbf{P}}_b = \mathbf{0}\ \forall t$. Hence, $\ddot{\mathbf{H}} = \dot{\mathbf{H}} = \mathbf{0}\ \forall t$. Substituting $\dot{\mathbf{P}}_b = \mathbf{0}$ and $\dot{\mathbf{H}} = \mathbf{0}$ in (5.27), we obtain $\dot{\gamma} = 0$.

2) γ is invariant under $\mathbf{U}_s$.

We will show that, under $\mathbf{U}_s$, it holds that

$$\mathrm{tr}(\mathbf{P}_b^\intercal \dot{\mathbf{P}}_b) = \mathrm{tr}(\mathbf{P}_{db}^\intercal \dot{\mathbf{H}}^\intercal \mathbf{P}_b) \ , \tag{5.28}$$

which implies, from (5.27), the invariance of γ. Notice

$$\ddot{\mathbf{P}}_b = \mathbf{U}_s \mathbf{K}_b = -k_{1s}(1 - s_d/s)\mathbf{H}\mathbf{P}_{db} - k_{2s}\dot{\mathbf{P}}_b \ . \tag{5.29}$$

We can use this to compute

$$\begin{aligned}\ddot{h}_1 &= \frac{\mathrm{tr}\Big((-k_{1s}(1 - s_d/s)\mathbf{H}\mathbf{P}_{db} - k_{2s}\dot{\mathbf{P}}_b)\mathbf{P}_{db}^\intercal\Big)}{\mathrm{tr}(\mathbf{P}_{db}\mathbf{P}_{db}^\intercal)} \\ &= -k_{1s}(1 - s_d/s)h_1 - k_{2s}\dot{h}_1 \ ,\end{aligned} \tag{5.30}$$

where we used $h_1 = \mathrm{tr}(\mathbf{H}\mathbf{P}_{db}\mathbf{P}_{db}^\intercal)/\mathrm{tr}(\mathbf{P}_{db}\mathbf{P}_{db}^\intercal)$ [11]. Since an analogous equation holds for $\ddot{h}_2$, we get

$$\ddot{\mathbf{h}} + k_{2s}\dot{\mathbf{h}} + k_{1s}(1 - s_d/\|\mathbf{h}\|)\mathbf{h} = \mathbf{0} \ . \tag{5.31}$$

where we defined $\mathbf{h} = [h_1, h_2]^\intercal$, with $\|\mathbf{h}\| = s$. Starting from rest ($\dot{\mathbf{h}}(t = 0) = \mathbf{0}$), we see that the dynamics of this equation must remain proportional to $\mathbf{h}$; i.e., $\mathbf{h}(t) = \kappa(t)\mathbf{h}(t = 0)$ with $\kappa(t)$ being a scalar. Note that $\kappa(t) \neq 0$ because $\kappa(t) = 0$ would imply $s(t) = 0$, which is ruled out by Assumption 5.1. We can directly write for $\mathbf{H}$:

$$\mathbf{H}(t) = \kappa(t)\mathbf{H}(0) \ . \tag{5.32}$$

where $\mathbf{H}(0)$ is $\mathbf{H}$ at time zero. By reorganizing (5.29) and substituting (5.32), we get

$$\ddot{\mathbf{P}}_b + k_{2s}\dot{\mathbf{P}}_b + k_{1s}(1 - s_d/s)\kappa(t)\mathbf{H}(0)\mathbf{P}_{db} = \mathbf{0}. \tag{5.33}$$

Defining a constant matrix $\mathbf{P}_{dbo} = \mathbf{H}(0)\mathbf{P}_{db}$, we express the above as a first order differential equation in $\dot{\mathbf{P}}_b$:

$$\ddot{\mathbf{P}}_b + k_{2s}\dot{\mathbf{P}}_b + \kappa_1(t)\mathbf{P}_{dbo} = \mathbf{0} \ , \tag{5.34}$$

for some scalar $\kappa_1(t)$. Starting from rest ($\dot{\mathbf{P}}_b(t = 0) = \mathbf{0}$), the solution to this equation has the form

$$\dot{\mathbf{P}}_b = \kappa_2(t)\mathbf{P}_{dbo} = \kappa_2(t)\mathbf{H}(0)\mathbf{P}_{db} = \mu(t)\mathbf{H}\mathbf{P}_{db}, \tag{5.35}$$

with $\kappa_2(t)$, $\mu(t)$ scalars such that $\mu(t) = \kappa_2(t)/\kappa(t)$. Hence

$$\dot{h}_1 = \frac{\mathrm{tr}(\dot{\mathbf{P}}_b\mathbf{P}_{db}^\intercal)}{\mathrm{tr}(\mathbf{P}_{db}\mathbf{P}_{db}^\intercal)} = \frac{\mu(t)\,\mathrm{tr}(\mathbf{H}\mathbf{P}_{db}\mathbf{P}_{db}^\intercal)}{\mathrm{tr}(\mathbf{P}_{db}\mathbf{P}_{db}^\intercal)} = \mu(t)h_1 \ , \tag{5.36}$$

where we have used $\mathrm{tr}(\mathbf{H}\mathbf{P}_{db}\mathbf{P}_{db}^{\intercal}) = \mathrm{tr}(\mathbf{P}_b\mathbf{P}_{db}^{\intercal})$, which can be deduced from (5.24). Note $\dot{h}_2 = \mu(t)h_2$ holds too. Using this, (5.35) and (5.24) in (5.28) we get:

$$\begin{aligned}\mathrm{tr}(\mathbf{P}_b^{\intercal}\dot{\mathbf{P}}_b) &= \mu(t)\,\mathrm{tr}(\mathbf{P}_b^{\intercal}(\mathbf{H}\mathbf{P}_{db}))\\ &= \mu(t)h_1\,\mathrm{tr}(\mathbf{P}_b^{\intercal}\mathbf{P}_{db}) + \mu(t)h_2\,\mathrm{tr}(\mathbf{P}_b^{\intercal}\mathbf{S}\mathbf{P}_{db})\\ &= \dot{h}_1\,\mathrm{tr}(\mathbf{P}_b^{\intercal}\mathbf{P}_{db}) + \dot{h}_2\,\mathrm{tr}(\mathbf{P}_b^{\intercal}\mathbf{S}\mathbf{P}_{db})\\ &= \mathrm{tr}(\mathbf{P}_b^{\intercal}\dot{\mathbf{H}}\mathbf{P}_{db}) = \mathrm{tr}(\mathbf{P}_{db}^{\intercal}\dot{\mathbf{H}}^{\intercal}\mathbf{P}_b)\ . \end{aligned} \tag{5.37}$$

Therefore, (5.28) holds. Hence, γ is invariant under $\mathbf{U}_s$.

3) γ is invariant under $\mathbf{U}_\theta$.

By recovering once again the expression for $\ddot{h}_1$ and substituting $\ddot{\mathbf{P}}_b = \mathbf{U}_\theta\mathbf{K}_b$, we get

$$\begin{aligned}\ddot{h}_1 &= \Big(\mathrm{tr}(\mathbf{P}_{db}\mathbf{P}_{db}^{\intercal})\Big)^{-1}\mathrm{tr}\Big((-k_{1\theta}e_\theta\mathbf{S}\mathbf{H}\mathbf{P}_{db} - k_{2\theta}\dot{\mathbf{P}}_b)\mathbf{P}_{db}^{\intercal}\Big)\\ &= -k_{1\theta}e_\theta\Big(\mathrm{tr}(\mathbf{P}_{db}\mathbf{P}_{db}^{\intercal})\Big)^{-1}\mathrm{tr}\Big(\mathbf{S}(h_1\mathbf{P}_{db} + h_2\mathbf{S}\mathbf{P}_{db})\mathbf{P}_{db}^{\intercal}\Big)\\ &- k_{2\theta}\dot{h}_1 = k_{1\theta}e_\theta h_2 - k_{2\theta}\dot{h}_1\ , \end{aligned} \tag{5.38}$$

where we used that $\mathbf{S}\mathbf{S} = -\mathbf{I}_{2\times 2}$. Similarly, we find $\ddot{h}_2 = -k_{1\theta}e_\theta h_1 - k_{2\theta}\dot{h}_2$. Then, we can obtain:

$$\ddot{\mathbf{H}} + k_{2\theta}\dot{\mathbf{H}} + k_{1\theta}e_\theta\mathbf{S}\mathbf{H} = \mathbf{0}\ . \tag{5.39}$$

On the other hand, we can rewrite $\mathbf{U}_\theta\mathbf{K}_b$ as

$$\ddot{\mathbf{P}}_b + k_{2\theta}\dot{\mathbf{P}}_b + k_{1\theta}e_\theta\mathbf{S}\mathbf{H}\mathbf{P}_{db} = \mathbf{0}\ . \tag{5.40}$$

If we post-multiply (5.39) by $\mathbf{P}_{db}$ and we subtract the result from (5.40), we obtain

$$(\ddot{\mathbf{P}}_b - \ddot{\mathbf{H}}\mathbf{P}_{db}) + k_{2\theta}(\dot{\mathbf{P}}_b - \dot{\mathbf{H}}\mathbf{P}_{db}) = \mathbf{0}\ . \tag{5.41}$$

Assuming the system initially at rest, i.e., $\dot{\mathbf{P}}_b - \dot{\mathbf{H}}\mathbf{P}_{db} = \mathbf{0}$ at $t = 0$, clearly $\ddot{\mathbf{P}}_b = \ddot{\mathbf{H}}\mathbf{P}_{db}$ and $\dot{\mathbf{P}}_b = \dot{\mathbf{H}}\mathbf{P}_{db}$ $\forall t$. Then, the equality (5.28) is satisfied and (5.27) equals zero. Hence, $\dot{\gamma} = 0$. This indicates that γ is invariant under $\mathbf{U}_\theta$. □

Proposition 5.2. *The formation variable* $\mathbf{g}$ *is invariant under* $\mathbf{U}_\gamma$, $\mathbf{U}_s$ *and* $\mathbf{U}_\theta$.

Proof. Notice that every addend in $\mathbf{U}_\gamma$, $\mathbf{U}_s$ and $\mathbf{U}_\theta$ can be expressed as $\mathbf{U}_i\mathbf{K}_b$. Therefore, for every such addend the dynamics of $\mathbf{g}$ is $\ddot{\mathbf{g}} = \frac{1}{N}\ddot{\mathbf{P}}\ \mathbf{1}_N = \frac{1}{N}\mathbf{U}_i\ \mathbf{K}_b\mathbf{1}_N$. Since $\mathbf{K}_b\mathbf{1}_N = \mathbf{0}$, $\ddot{\mathbf{g}} = \mathbf{0}$. Assuming the system is initially at rest (i.e., $\dot{\mathbf{g}}(t = 0) = \mathbf{0}$), we conclude $\dot{\mathbf{g}} = \mathbf{0}$ $\forall t$. □

Proposition 5.3. *The formation variable* θ *is invariant under* $\mathbf{U}_\gamma$, $\mathbf{U}_g$ *and* $\mathbf{U}_s$.

Proof. We start with $\mathbf{U}_\gamma$. As $\theta = \text{atan2}(h_2, h_1)$, we study the dynamics of h_1 and h_2. Under $\mathbf{U}_\gamma$, $\ddot{\mathbf{P}} = \mathbf{U}_\gamma$. From the properties of $\mathbf{K}_b$, we have $\mathbf{U}_\gamma \mathbf{K}_b = \mathbf{U}_\gamma$ and therefore $\ddot{\mathbf{P}}_b = \ddot{\mathbf{P}}\mathbf{K}_b = \mathbf{U}_\gamma \mathbf{K}_b = \mathbf{U}_\gamma$. We can then write:

$$\begin{aligned}
\ddot{h}_1 =& \Big(\text{tr}(\mathbf{P}_{db}\mathbf{P}_{db}^\mathsf{T})\Big)^{-1} \text{tr}(\ddot{\mathbf{P}}_b\mathbf{P}_{db}^\mathsf{T}) = \Big(\text{tr}(\mathbf{P}_{db}\mathbf{P}_{db}^\mathsf{T})\Big)^{-1} \text{tr}(\mathbf{U}_\gamma\mathbf{P}_{db}^\mathsf{T}) \\
=& \Big(\text{tr}(\mathbf{P}_{db}\mathbf{P}_{db}^\mathsf{T})\Big)^{-1} \text{tr}\Big(\alpha_H(k_{1H}(\mathbf{H}\mathbf{P}_{db}\mathbf{P}_{db}^\mathsf{T} - \mathbf{P}_b\mathbf{P}_{db}^\mathsf{T}) \\
&+ k_{2H}(\dot{\mathbf{H}}\mathbf{P}_{db}\mathbf{P}_{db}^\mathsf{T} - \dot{\mathbf{P}}_b\mathbf{P}_{db}^\mathsf{T})) \\
&+ \alpha_G(k_{1G}(\mathbf{P}_b\mathbf{P}_{db}^\mathsf{T}(\mathbf{P}_{db}\mathbf{P}_{db}^\mathsf{T})^{-1}\mathbf{P}_{db}\mathbf{P}_{db}^\mathsf{T} - \mathbf{P}_b\mathbf{P}_{db}^\mathsf{T}) \\
&+ k_{2G}(\dot{\mathbf{P}}_b\mathbf{P}_{db}^\mathsf{T}(\mathbf{P}_{db}\mathbf{P}_{db}^\mathsf{T})^{-1}\mathbf{P}_{db}\mathbf{P}_{db}^\mathsf{T} - \dot{\mathbf{P}}_b\mathbf{P}_{db}^\mathsf{T}))\Big). \qquad (5.42)
\end{aligned}$$

Since $(\mathbf{P}_{db}\mathbf{P}_{db}^\mathsf{T})^{-1}\mathbf{P}_{db}\mathbf{P}_{db}^\mathsf{T} = \mathbf{I}_{2\times 2}$, the term multiplied by α_G is zero. Besides, it holds that $\text{tr}(\mathbf{P}_b\mathbf{P}_{db}^\mathsf{T}) = \text{tr}(\mathbf{H}\mathbf{P}_{db}\mathbf{P}_{db}^\mathsf{T})$, as noted above; and, taking the time derivative, $\text{tr}(\dot{\mathbf{P}}_b\mathbf{P}_{db}^\mathsf{T}) = \text{tr}(\dot{\mathbf{H}}\mathbf{P}_{db}\mathbf{P}_{db}^\mathsf{T})$. If we apply these relations in (5.42), we get $\ddot{h}_1 = 0$. By following the same procedure for $\ddot{h}_2$, we obtain $\ddot{h}_2 = 0$, which yields $\ddot{\mathbf{H}} = \mathbf{0}$. Assuming that the system is at rest at $t = 0$, i.e., $\dot{\mathbf{P}}(t = 0) = \mathbf{0}$, we determine that $\dot{\mathbf{H}} = \mathbf{0}$. Hence, θ is invariant under $\mathbf{U}_\gamma$.

Under $\mathbf{U}_g$, $\ddot{\mathbf{P}}_b = \mathbf{U}_g\mathbf{K}_b = -k_{1g}\mathbf{e}_g\mathbf{1}_N^\mathsf{T}\mathbf{K}_b - k_{2g}\dot{\mathbf{P}}_b$. As $\mathbf{1}_N^\mathsf{T}\mathbf{K}_b = \mathbf{0}$, and assuming $\dot{\mathbf{P}}_b(t = 0) = \mathbf{0}$, we have $\ddot{\mathbf{P}}_b = \dot{\mathbf{P}}_b = \mathbf{0}\ \forall t$. Hence, $\dot{\mathbf{H}} = \mathbf{0}\ \forall t$, and θ is invariant.

Under $\mathbf{U}_s$, we know from (5.32) that $\mathbf{H}(t) = \kappa(t)\mathbf{H}(0)$ with $\kappa(t)$ being a scalar. Therefore, h_2/h_1 is constant. If $h_1 = 0$ initially, then $h_1 = 0\ \forall t$, and h_2/h_1 is always of infinite magnitude. In conclusion, θ is invariant. □

Proposition 5.4. *The formation variable s is invariant under $\mathbf{U}_\gamma$ and $\mathbf{U}_g$. It is not invariant under $\mathbf{U}_\theta$.*

Proof. In Proposition 5.3 we concluded that $\dot{\mathbf{H}} = \mathbf{0}$ under $\mathbf{U}_\gamma$ and under $\mathbf{U}_g$. Hence, $s = ||\mathbf{H}||_2$ is invariant under $\mathbf{U}_\gamma$ and under $\mathbf{U}_g$.

Under $\mathbf{U}_\theta$, the third term of (5.39), which includes $\mathbf{SH}$, generates the rotation movement. With dynamics of this form, $s = ||\mathbf{H}||_2$ is not constant in the general case. □

Remark 5.1. *The strong invariance properties that we have proven for γ, $\mathbf{g}$ and θ are favorable for achieving a fully uncoupled control. The scale variable s is not invariant under $\mathbf{U}_\theta$ (Proposition 5.4), which creates an undesired coupling that was already expected from the dynamics (5.19), (5.20). Despite this, through the use of the term $\mathbf{U}_u$ we will achieve fully uncoupled control of all the variables (γ, $\mathbf{g}$, θ and s), as shown in the next section.*

5.5 Stability and convergence under $\mathbf{U}$

The following is our main formal result, which establishes the stability and convergence of the proposed controller, and provides the dynamics of the variables.

Theorem 5.1. *The multirobot system under the action of* $\mathbf{U}$ (5.22) *is stable and the robot positions* $\mathbf{P}$ *converge asymptotically to the target configuration* $\mathbf{P}_T$. *Moreover, each variable (*γ, $\mathbf{g}$, θ *and* s*) evolves according to linear dynamics uncoupled from the dynamics of the other variables.*

Proof. We will first compute the dynamics imposed by the control law $\mathbf{U}$ (5.22) for $\mathbf{P}$, h_1 and h_2, and then we will use this to obtain the dynamics of the formation variables. We start by computing $\ddot{\mathbf{P}}$:

$$\begin{aligned}\ddot{\mathbf{P}} = \,&\alpha_H k_{1H}(\mathbf{H}\mathbf{P}_{db} - \mathbf{P}_b) + \alpha_H k_{2H}(\dot{\mathbf{H}}\mathbf{P}_{db} - \dot{\mathbf{P}}_b)\\ &+ \alpha_G k_{1G}(\mathbf{G}\mathbf{P}_{db} - \mathbf{P}_b) + \alpha_G k_{2G}(\dot{\mathbf{G}}\mathbf{P}_{db} - \dot{\mathbf{P}}_b)\\ &- k_{1g}\mathbf{e}_g\mathbf{1}_N^{\mathsf{T}} - k_{2g}\dot{\mathbf{P}} - k_{1s}e_s(1/s)\mathbf{H}\mathbf{P}_{db} - k_{2s}\dot{\mathbf{P}}_b\\ &- k_{1\theta}e_\theta\mathbf{S}\mathbf{H}\mathbf{P}_{db} - k_{2\theta}\dot{\mathbf{P}}_b - \dot{\theta}^2\mathbf{H}\mathbf{P}_{db} + \frac{2\dot{s}\dot{\theta}}{s}\mathbf{S}\mathbf{H}\mathbf{P}_{db}.\end{aligned} \tag{5.43}$$

Let us define $k_{2f} = k_{2g} + k_{2s} + k_{2\theta}$ and

$$\eta = k_{1s}\frac{e_s}{s} + \dot{\theta}^2, \quad \rho = k_{1\theta}e_\theta - \frac{2\dot{s}\dot{\theta}}{s}. \tag{5.44}$$

We then have these expressions for $\ddot{h}_1$ and $\ddot{h}_2$:

$$\begin{aligned}\ddot{h}_1 = &- k_{2g}\dot{h}_1 - k_{1s}(e_s/s)h_1 - k_{2s}\dot{h}_1 + k_{1\theta}e_\theta h_2 - k_{2\theta}\dot{h}_1\\ &- \dot{\theta}^2 h_1 - \frac{2\dot{s}\dot{\theta}}{s}h_2 = -k_{2f}\dot{h}_1 - \eta h_1 + \rho h_2.\end{aligned} \tag{5.45}$$

$$\begin{aligned}\ddot{h}_2 = &- k_{2g}\dot{h}_2 - k_{1s}(e_s/s)h_2 - k_{2s}\dot{h}_2 - k_{1\theta}e_\theta h_1 - k_{2\theta}\dot{h}_2\\ &- \dot{\theta}^2 h_2 + \frac{2\dot{s}\dot{\theta}}{s}h_1 = -k_{2f}\dot{h}_2 - \eta h_2 - \rho h_1.\end{aligned} \tag{5.46}$$

Recall that the control term $\mathbf{U}_\gamma$ produces $\ddot{h}_1 = \ddot{h}_2 = 0$, which is why it does not appear in (5.45) and (5.46). We can now obtain the error dynamics of each variable.

1) Error dynamics of γ. Note γ itself is an error variable with respect to its desired value $\gamma = \gamma_d = 0$. To analyze γ we will compute the dynamics of the variables $\mathbf{E}_\gamma$ and $\mathbf{E}_{GH}$. To this end, we first write the expression for $\ddot{\mathbf{P}}_b$ from (5.43), using that $\mathbf{1}_N^{\mathsf{T}}\mathbf{K}_b = \mathbf{0}$:

$$\begin{aligned}\ddot{\mathbf{P}}_b = \,&\alpha_H k_{1H}(\mathbf{H}\mathbf{P}_{db} - \mathbf{P}_b) + \alpha_H k_{2H}(\dot{\mathbf{H}}\mathbf{P}_{db} - \dot{\mathbf{P}}_b)\\ &+ \alpha_G k_{1G}(\mathbf{G}\mathbf{P}_{db} - \mathbf{P}_b) + \alpha_G k_{2G}(\dot{\mathbf{G}}\mathbf{P}_{db} - \dot{\mathbf{P}}_b)\\ &- k_{2f}\dot{\mathbf{P}}_b - \eta\mathbf{H}\mathbf{P}_{db} - \rho\mathbf{S}\mathbf{H}\mathbf{P}_{db}.\end{aligned} \tag{5.47}$$

We compute from (5.45) and (5.46) the dynamics of $\mathbf{H}$:

$$\ddot{\mathbf{H}} = -k_{2f}\dot{\mathbf{H}} - \eta\mathbf{H} - \rho\mathbf{S}\mathbf{H}. \tag{5.48}$$

For $\mathbf{G}$, from (5.47) and using $\mathbf{P}_{db}\mathbf{P}_{db}^{+} = \mathbf{I}_{2\times 2}$ we obtain

$$\ddot{\mathbf{G}} = \ddot{\mathbf{P}}_b\mathbf{P}_{db}^{+} = -k_{2f}\dot{\mathbf{G}} - \alpha_H k_{1H}(\mathbf{G}-\mathbf{H}) - \alpha_H k_{2H}(\dot{\mathbf{G}}-\dot{\mathbf{H}}) - \eta\mathbf{H} - \rho\mathbf{S}\mathbf{H}. \tag{5.49}$$

Notice that

$$\ddot{\mathbf{G}} - \ddot{\mathbf{H}} = -\alpha_H k_{1H}(\mathbf{G}-\mathbf{H}) - (\alpha_H k_{2H} + k_{2f})(\dot{\mathbf{G}}-\dot{\mathbf{H}}). \tag{5.50}$$

Now, using $\ddot{\mathbf{P}}_b - \ddot{\mathbf{H}}\mathbf{P}_{db} = \ddot{\mathbf{P}}_b - \ddot{\mathbf{G}}\mathbf{P}_{db} + (\ddot{\mathbf{G}} - \ddot{\mathbf{H}})\mathbf{P}_{db}$, we can directly get, substituting equations (5.47) to (5.50):

$$\begin{aligned}\ddot{\mathbf{E}}_\gamma = \ddot{\mathbf{P}}_b - \ddot{\mathbf{H}}\mathbf{P}_{db} &= -(\alpha_H k_{1H} + \alpha_G k_{1G})\mathbf{E}_\gamma - (\alpha_H k_{2H} + \alpha_G k_{2G} + k_{2f})\dot{\mathbf{E}}_\gamma \\ &+ \alpha_G k_{1G}\mathbf{E}_{GH} + \alpha_G k_{2G}\dot{\mathbf{E}}_{GH},\end{aligned} \tag{5.51}$$

$$\ddot{\mathbf{E}}_{GH} = \ddot{\mathbf{G}}\mathbf{P}_{db} - \ddot{\mathbf{H}}\mathbf{P}_{db} = -\alpha_H k_{1H}\mathbf{E}_{GH} - (\alpha_H k_{2H} + k_{2f})\dot{\mathbf{E}}_{GH}.$$

Notice that this forms a linear system for each of the $2N$ components (i.e., position coordinates) of $\mathbf{E}_\gamma$ and $\mathbf{E}_{GH}$. Every one of these systems has the same dynamics. Therefore, it suffices to study one of them. Let us take an arbitrary $i \in \{1, ..., 2N\}$ and call $e_\gamma \in \mathbb{R}$ and $e_{GH} \in \mathbb{R}$ the components of $\mathbf{E}_\gamma$ and $\mathbf{E}_{GH}$, respectively, corresponding to that i. We can then write

$$\begin{bmatrix}\dot{e}_\gamma \\ \ddot{e}_\gamma \\ \dot{e}_{GH} \\ \ddot{e}_{GH}\end{bmatrix} = \underbrace{\begin{bmatrix}0 & 1 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} \\ 0 & 0 & 0 & 1 \\ 0 & 0 & a_{43} & a_{44}\end{bmatrix}}_{\mathbf{A}_e}\begin{bmatrix}e_\gamma \\ \dot{e}_\gamma \\ e_{GH} \\ \dot{e}_{GH}\end{bmatrix}, \tag{5.52}$$

where $a_{21} = -(\alpha_H k_{1H} + \alpha_G k_{1G})$, $a_{22} = -(\alpha_H k_{2H} + \alpha_G k_{2G} + k_{2f})$, $a_{23} = \alpha_G k_{1G}$, $a_{24} = \alpha_G k_{2G}$, $a_{43} = -\alpha_H k_{1H}$, $a_{44} = -(\alpha_H k_{2H} + k_{2f})$. We compute the characteristic polynomial of $\mathbf{A}_e$:

$$|\lambda\mathbf{I} - \mathbf{A}_e| = (\lambda^2 - a_{22}\lambda - a_{21})(\lambda^2 - a_{44}\lambda - a_{43}). \tag{5.53}$$

As a_{22}, a_{21}, a_{44} and a_{43} are all strictly negative, from the Routh-Hurwitz criterion the eigenvalues of $\mathbf{A}_e$ have negative real parts, and hence the system is stable. Therefore, $\mathbf{E}_\gamma$, $\dot{\mathbf{E}}_\gamma$, $\mathbf{E}_{GH}$ and $\dot{\mathbf{E}}_{GH}$ are all bounded and they converge to zero asymptotically. Notice, then, that $\gamma = (1/2)\mathrm{tr}(\mathbf{E}_\gamma^\intercal\mathbf{E}_\gamma)$ is bounded and converges to zero asymptotically. Moreover, the dynamics of $\mathbf{E}_\gamma$ are fully determined by the initial configuration and matrix $\mathbf{A}_e$: therefore, γ does not depend on the other variables being controlled ($\mathbf{g}$, s, θ).

2) Error dynamics of $\mathbf{g}$. We substitute (5.43) in $\ddot{\mathbf{g}} = (1/N)\ddot{\mathbf{P}}\mathbf{1}_N$. Note that every addend ending in a b in (5.43) is being post-multiplied by $\mathbf{K}_b$. Then, since $\mathbf{K}_b\mathbf{1}_N = \mathbf{0}$ and $\mathbf{1}_N^\intercal\mathbf{1}_N = N$, we directly find: $\ddot{\mathbf{g}} = -k_{1g}\mathbf{e}_g - k_{2g}\dot{\mathbf{g}}$. Therefore, we have:

$$\ddot{\mathbf{e}}_g = -k_{1g}\mathbf{e}_g - k_{2g}\dot{\mathbf{e}}_g. \tag{5.54}$$

We can define $\mathbf{e}_g = [e_{gx}, e_{gy}]^\intercal$, and then for each component j ($j \in \{x, y\}$) we have a linear system $\dot{\mathbf{x}}_{egj} = \mathbf{A}_g \mathbf{x}_{egj}$ on the state $\mathbf{x}_{egj} = [e_{gj}, \dot{e}_{gj}]^\intercal$. The characteristic polynomial of $\mathbf{A}_g$ is $\lambda^2 + k_{2g}\lambda + k_{1g}$. As k_{1g} and k_{2g} are positive, from the Routh-Hurwitz criterion $\mathbf{e}_g$ and $\dot{\mathbf{e}}_g$ are bounded and stable, converging to zero over time. Hence, the centroid $\mathbf{g}$ converges to the desired one, $\mathbf{g}_d$.

3) Error dynamics of s. Substituting (5.45) and (5.46) in the second equation of (5.19) and applying $h_1\dot{h}_1 + h_2\dot{h}_2 = \dot{s}s$ (5.19) and $h_1^2 + h_2^2 = s^2$, we find $\ddot{s} = -k_{1s}e_s - k_{2f}\dot{s}$, i.e.:

$$\ddot{e}_s = -k_{1s}e_s - k_{2f}\dot{e}_s. \tag{5.55}$$

We define a linear system $\dot{\mathbf{x}}_{es} = \mathbf{A}_s \mathbf{x}_{es}$ on the state $\mathbf{x}_{es} = [e_s, \dot{e}_s]^\intercal$. The characteristic polynomial of $\mathbf{A}_s$ is $\lambda^2 + k_{2f}\lambda + k_{1s}$. As k_{1s} and k_{2f} are positive, from the Routh-Hurwitz criterion e_s and $\dot{e}_s$ are bounded and stable, converging to zero over time. Hence, s converges to the desired scale, s_d.

4) Error dynamics of θ. Substituting (5.45) and (5.46) in the second equation of (5.20) and applying $h_1\dot{h}_2 - h_2\dot{h}_1 = \dot{\theta}s^2$ (5.20) and $h_1^2 + h_2^2 = s^2$, we find $\ddot{\theta} = -k_{1\theta}e_\theta - k_{2f}\dot{\theta}$, i.e.:

$$\ddot{e}_\theta = -k_{1\theta}e_\theta - k_{2f}\dot{e}_\theta. \tag{5.56}$$

We now have a linear system $\dot{\mathbf{x}}_{e\theta} = \mathbf{A}_\theta \mathbf{x}_{e\theta}$ on the state $\mathbf{x}_{e\theta} = [e_\theta, \dot{e}_\theta]^\intercal$. The characteristic polynomial of $\mathbf{A}_\theta$ is $\lambda^2 + k_{2f}\lambda + k_{1\theta}$. As $k_{1\theta}$ and k_{2f} are positive, from the Routh-Hurwitz criterion e_θ and $\dot{e}_\theta$ are bounded and stable, converging to zero over time. Thus, the angle θ converges to the desired one, θ_d.

Now, from **2)**, **3)** and **4)**, it is clear that the evolutions of $\mathbf{g}$, s and θ are determined by the initial configuration and the matrices $\mathbf{A}_g$, $\mathbf{A}_s$, and $\mathbf{A}_\theta$, respectively, which depend only on the chosen control gains. Hence, during the task each variable evolves independently from the others.

Convergence to the target configuration. Notice from the dynamics above that $||\mathbf{P}_b - \mathbf{H}\mathbf{P}_{db}||$ and $||\dot{\mathbf{P}}_b - \dot{\mathbf{H}}\mathbf{P}_{db}||$ are bounded. In addition, s, θ, $\dot{s}$ and $\dot{\theta}$, which represent the value and dynamics of the norm and angle of $\mathbf{H}$, are also bounded. This implies that $||\mathbf{P}_b||$ and $||\dot{\mathbf{P}}_b||$ are bounded. Since $||\mathbf{g}||$ and $||\dot{\mathbf{g}}||$ are also bounded, we infer that $||\mathbf{P}||$ and $||\dot{\mathbf{P}}||$ are bounded. The convergence is to a static configuration, i.e., $\dot{\mathbf{P}} = \mathbf{0}$. This is because $\dot{\mathbf{H}}$ converges to zero (as $\dot{s}$ and $\dot{\theta}$ converge to zero), and therefore, $\dot{\mathbf{P}}_b = \dot{\mathbf{H}}\mathbf{P}_{db}$ converges to zero. Given that $\dot{\mathbf{g}}$ also converges to zero, $\dot{\mathbf{P}}$ converges to $\mathbf{0}$.

Recall the target configuration is $\mathbf{P}_T = s_d\mathbf{R}_d(\theta_d)\mathbf{P}_{db} + \mathbf{g}_d\mathbf{1}_N^\intercal$. Let us use a subscript c to denote the configuration that the system converges to, and its variables. This configuration satisfies $\mathbf{P}_c\mathbf{K}_b = \mathbf{H}_c\mathbf{P}_{db}$ (due to $\gamma = 0$), i.e., $\mathbf{P}_c = s_c\mathbf{R}_c\mathbf{P}_{db} + \mathbf{g}_c\mathbf{1}_N^\intercal$. As $e_s = 0$ and $e_\theta = 0$, we have $s_c = s_d$ and $\theta_c = \theta_d$. As $\mathbf{e}_g = \mathbf{0}$, we have $\mathbf{g}_c = \mathbf{g}_d$. Therefore, $\mathbf{P}_c = \mathbf{P}_T$. □

5.6 Safe control with CBFs

5.6.1 Collision avoidance

Due to the fact that the previous controller does not take into account explicitly collision avoidance, we need an additional strategy to complement the formation controller $\mathbf{U}$. Collisions may occur: between two agents (agent-to-agent), between an agent and an obstacle (agent-to-obstacle) and between the object and an obstacle (object-to-obstacle). In contrast to other multirobot systems, where the agents are not linked by a solid structure, collision avoidance is more challenging in our case: the robots are grasping the object, and their maneuvers must always respect the admissible deformation states of the object. Control barrier functions (CBFs) provide a robust, flexible and minimally invasive solution to this issue. We adapt and extend the centralized formulation proposed by Wang *et al.* [132], for collision avoidance in a team of robots that are assigned different position goals. Our system can be written in the affine form

$$\dot{\mathbf{x}}_{ij} = f(\mathbf{x}_{ij}) + g(\mathbf{x}_{ij})\mathbf{u}_{ij} \ , \tag{5.57}$$

where $\mathbf{x}_{ij} = [\mathbf{p}_{ij}, \dot{\mathbf{p}}_{ij}]^\intercal$, $\mathbf{p}_{ij} = \mathbf{p}_i - \mathbf{q}_j$, $\mathbf{u}_{ij} = \mathbf{u}_i - \mathbf{w}_j$ and $f(\mathbf{x}_{ij})$ and $g(\mathbf{x}_{ij})$ are locally Lipschitz continuous functions, which describe how the agents are coupled with each other via the controller. Note that $\mathbf{q}_j$ and $\mathbf{w}_j$ are the position and control input, respectively, of agent j, which can be a robot of the formation (and then $\mathbf{q}_j = \mathbf{p}_j$, $\mathbf{w}_j = \mathbf{u}_j$) or an obstacle. In the latter case, we must set $\mathbf{w}_j = \mathbf{0}$ (obstacles cannot be controlled). Then, we propose the following condition for collision avoidance, which restricts the distance between i and j to a minimum value:

$$\|\mathbf{p}_{ij}\| + \int_{t_0}^{t_f} \dot{p}_{ij}^{\perp}(t)dt \geq D_{ij}^{min} \ , \tag{5.58}$$

where

$$\dot{p}_{ij}^{\perp} = \frac{\mathbf{p}_{ij}^\intercal}{\|\mathbf{p}_{ij}\|}\dot{\mathbf{p}}_{ij} \tag{5.59}$$

is the normal component of the relative velocity $\dot{\mathbf{p}}_{ij}$,

$$t_f = \frac{\dot{p}_{ij}^{\perp}(t_f) - \dot{p}_{ij}^{\perp}(t_0)}{\alpha_i + \alpha_j} + t_0 \tag{5.60}$$

for $\dot{p}_{ij}^{\perp}(t_f) = 0$ is the time instant after having applied the maximum braking accelerations α_i and α_j ($\|\ddot{\mathbf{p}}_i\|_\infty = \alpha_i$), and D_{ij}^{min} is the minimum allowed distance between i and j. If j is an obstacle, $\alpha_j = 0$. After manipulation of the previous equations, and taking $t_0 = 0$, we get the candidate CBF for i and j

$$h_{ij} = 2(\alpha_i + \alpha_j)(\|\mathbf{p}_{ij}\| - D_{ij}^{min}) - \dot{p}_{ij}^{\perp 2} \ , \tag{5.61}$$

which is defined in $\mathbb{R}$ for both the safe and the unsafe regions. By definition [7], we know that $h_{ij}(\mathbf{x}_{ij})$ is a CBF if there exists an extended class K_∞ function $\varepsilon(h_{ij}(\mathbf{x}_{ij}))$ such that

$$\sup_{\mathbf{u}_{ij}}[L_f h_{ij}(\mathbf{x}_{ij}) + L_g h_{ij}(\mathbf{x}_{ij})\mathbf{u}_{ij}] \geq -\varepsilon(h_{ij}(\mathbf{x}_{ij})) \ , \tag{5.62}$$

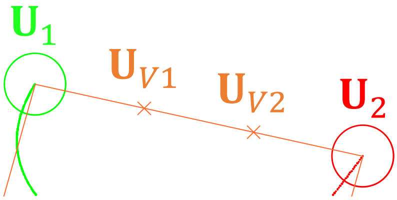


Figure 5.1: Two real agents (green and red circles) and two virtual agents (orange '×') in the formation edge that links $\mathbf{P}_1$ and $\mathbf{P}_2$. In this case, their control inputs are computed as $\mathbf{U}_{v1} = 2/3\,\mathbf{U}_1 + 1/3\,\mathbf{U}_2$ and $\mathbf{U}_{v2} = 1/3\,\mathbf{U}_1 + 2/3\,\mathbf{U}_2$.

where L represents the Lie derivative. By using this definition and substituting our candidate CBF (5.61), we obtain a set of linear constraints with respect to $\mathbf{u}_{ij}$:

$$\mathbf{p}_{ij}^{\mathsf{T}}\dot{\mathbf{p}}_{ij}\frac{\mathbf{p}_{ij}^{\mathsf{T}}}{\|\mathbf{p}_{ij}\|}\mathbf{u}_{ij} \leq (\alpha_i + \alpha_j)\mathbf{p}_{ij}^{\mathsf{T}}\dot{\mathbf{p}}_{ij} + \frac{\|\mathbf{p}_{ij}\|}{2}\varepsilon(h_{ij}) \ . \tag{5.63}$$

These constraints guarantee that the system will maintain a minimal distance of D_{ij}^{min} between the center of agent i and the center of agent/obstacle j. This implies that agents and obstacles are circles with regard to collision avoidance. However, if $\|\mathbf{p}_{ab}\| \geq D_{ac}^{min} + D_{bc}^{min}$, where a and b are neighboring agents and c an obstacle, c could penetrate into the formation. This behavior is undesired in the present case, because it would result in an object-to-obstacle collision.

The strategy we propose to solve this issue consists in deploying a set of virtual agents $\mathbf{p}_{vk} = [p_{vkx}, p_{vky}]^{\mathsf{T}}$, $k = 1, ..., V$ over the contour edges of the formation polygon (Fig. 5.1). These virtual agents add new distance constraints with respect to the obstacles, and their virtual inputs are computed as the linear combination

$$\mathbf{u}_{vk} = (1-\nu)\mathbf{u}_i + \nu\,\mathbf{u}_j \ , \tag{5.64}$$

where $\nu = \|\mathbf{p}_{vk} - \mathbf{p}_i\|/\|\mathbf{p}_j - \mathbf{p}_i\|$ and $\mathbf{p}_{vk}$ is the position of the virtual agent in the edge $\overline{i\,j}$. This strategy preserves the desired safety distance D_{ij}^{min}. In addition, the computational cost of the system does not increase substantially, since no new control inputs are added (the virtual inputs are computed from the ones of the real agents).

5.6.2 Overstretching avoidance

Yet minimally invasive over $\mathbf{U}$, the collision avoidance system may separate two agents beyond the deformation limit of the carried object. We can write the condition to avoid overstretching the object as a restriction on the maximum distance between i and j:

$$\|\mathbf{p}_{ij}\| + \int_{t_0}^{t_f} \dot{p}_{ij}^{\perp}(t)dt \leq D_{ij}^{max} \ , \tag{5.65}$$

where

$$t_f = \frac{\dot{p}_{ij}^{\perp}(t_0)}{\alpha_i + \alpha_j} - t_0 \tag{5.66}$$

and D_{ij}^{max} is the maximum allowed separation between i and j. Note that this distance can be set differently for each pair of agents, so that more fragile parts of the object are constrained to a higher extent than the rest. The following CBF candidate integrates the previous equations as

$$h'_{ij} = 2(\alpha_i + \alpha_j)(D_{ij}^{max} - \|\mathbf{p}_{ij}\|) - \dot{p}_{ij}^{\perp 2} \ . \tag{5.67}$$

By substituting (5.67) in (5.62) we get the new set of constraints

$$\frac{\mathbf{p}_{ij}^{\intercal}}{\|\mathbf{p}_{ij}\|}\dot{\mathbf{p}}_{ij}\mathbf{p}_{ij}^{\intercal}\mathbf{U}_{ij} \le -(\alpha_i + \alpha_j)\mathbf{p}_{ij}^{\intercal}\dot{\mathbf{p}}_{ij} + \frac{\|\mathbf{p}_{ij}\|}{2}\varepsilon'(h'_{ij}) \ . \tag{5.68}$$

These constraints prevent stretching the object more than D_{ij}^{max} between every pair of agents i and j.

5.6.3 Quadratic programming-based controller

Finally, we introduce the previous constraints into a Quadratic Programming-based (QP) controller, which outputs the safe control inputs $\mathbf{U}_B = [\mathbf{u}_{B1}^{\intercal}, \mathbf{u}_{B2}^{\intercal}, ..., \mathbf{u}_{BN}^{\intercal}]^{\intercal} \in \mathbb{R}^{2N}$:

$$\begin{aligned}
&\text{Given} && \mathbf{U}, \mathbf{P}, \dot{\mathbf{P}}, D_{ij}^{min}, D_{ij}^{max}, \alpha_i, \alpha_j \\
&\underset{\mathbf{U}_B}{\text{minimize}} && \xi = \sum_{i=1}^{N} \|\mathbf{u}_{Bi} - \mathbf{u}_i\|_2^2 \\
&\text{subject to:} && \\
& && \mathbf{A}_{ij}\mathbf{U}_B \le b_{ij} \ , \forall i \ne j, \ i = 1, ..., N, \ j = 1, ..., N + M \ , \\
& && \mathbf{A}_{kj}\mathbf{U}_B \le b_{kj} \ , k = 1, ..., V, \ j = 1, ..., M \ , \\
& && \mathbf{A}'_{ij}\mathbf{U}_B \le b'_{ij} \ , \forall i \ne j, \ i = 1, ..., N, \ j = 1, ..., N \ , \\
& && \|\mathbf{u}_{Bi}\|_\infty \le \alpha_i, i = 1, ..., N
\end{aligned} \tag{5.69}$$

where M is the number of obstacles, $\mathbf{A}_{ij}$, $\mathbf{A}_{kj}$ and $\mathbf{A}'_{ij}$ are defined as

$$\mathbf{A}_{**} = [0, ..., \mathbf{p}_{**}^{\intercal}\dot{\mathbf{p}}_{**}\frac{\mathbf{p}_{**}^{\intercal}}{\|\mathbf{p}_{**}\|}, ..., -\mathbf{p}_{**}^{\intercal}\dot{\mathbf{p}}_{**}\frac{\mathbf{p}_{**}^{\intercal}}{\|\mathbf{p}_{**}\|}, ..., 0] \ , \tag{5.70}$$

being subscript $**$ the corresponding indexes and the first non-zero terms in the i_{th}/k_{th} position and the second in the j_{th} position,

$$b_{ij} = (\alpha_i + \alpha_j)\mathbf{p}_{ij}^{\mathsf{T}}\dot{\mathbf{p}}_{ij} + \frac{\|\mathbf{p}_{ij}\|}{2}\varepsilon(h_{ij}) \,, \tag{5.71}$$

$$b_{kj} = ((1-\nu)\alpha_i + \nu\,\alpha_j)\mathbf{p}_{kj}^{\mathsf{T}}\dot{\mathbf{p}}_{kj} + \frac{\|\mathbf{p}_{kj}\|}{2}\varepsilon(h_{kj}) \,, \tag{5.72}$$

$$b'_{ij} = -(\alpha_i + \alpha_j)\mathbf{p}_{ij}^{\mathsf{T}}\dot{\mathbf{p}}_{ij} + \frac{\|\mathbf{p}_{ij}\|}{2}\varepsilon'(h'_{ij}) \,. \tag{5.73}$$

The QP controller computes, for each agent, the closest control input $\mathbf{u}_{Bi}$ to the nominal $\mathbf{U}_i$ that satisfies the collision and overstretching avoidance constraints.

5.7 Implementation details

5.7.1 Design of control gains

As can be seen from the error dynamics presented above, we can control in an independent manner the specific time evolutions of $\mathbf{g}$, s and θ. This is another advantage of our new control law relative to the one in [57].

Commonly, it can be desirable to avoid overshooting; to this end, we can choose the gains in the overdamped region, i.e., $k_{2j} \geq 2\sqrt{k_{1j}}$, for $j \in \{H, G, g\}$. In the case of s and θ, their dynamics share the coefficient k_{2f} (see (5.55) and (5.56)); then, to avoid overshooting, one option is to fix k_{2f} and then define $k_{1s} \leq k_{2f}^2/4$, $k_{1\theta} \leq k_{2f}^2/4$. The choice of α_H and α_G can be made on the basis of how large the deformations are during execution. When large deformations are required, a large α_G is useful as it can keep these deformation close to affine. Note, finally, that the actual time evolutions of the variables will also be determined by the safety constraints imposed via the CBFs.

5.7.2 Handling of measurement noise

In implementation scenarios with significant measurement noise, the accuracy of the control term $\mathbf{U}_u$ estimated from the measurements will be particularly affected. Some practical actions that can be taken to reduce this effect are:

- The term $-\dot{\theta}^2$ in $\mathbf{U}_u$ always has the same sign and hence the noise associated with it can create a steady-state negative offset in e_s. A way to reduce this effect without significantly impacting the overall control performance is to dampen this term when some measure of error, d_e, is below a small chosen value, $d_n < 1$; e.g., by using:

$$\mathbf{U}_u = -d_e\dot{\theta}^2\mathbf{H}\mathbf{P}_{db} + \frac{2\dot{s}\dot{\theta}}{s}\mathbf{S}\mathbf{H}\mathbf{P}_{db}, \text{ if } d_e < d_n. \tag{5.74}$$

A suitable choice is, e.g., $d_e = \kappa_\theta |e_\theta|$ for a $\kappa_\theta \geq 0$.

- Choosing $\theta_d = 0$ can help as it may lower the magnitude of noise in practice. Note that it is always possible for us to choose $\theta_d = 0$, by defining $\mathbf{P}_d$ appropriately.
- Filtering measurements or estimations over a time window, e.g. by means of the average value, can also decrease the effect of noise.

The adjustments above should be made while ensuring that the team can still react fast enough to satisfy the safety constraints.

5.8 Experimental validation

5.8.1 Full controller simulation tests

We start by testing $\mathbf{U}$, i.e. the full formation controller with the uncoupling term $\mathbf{U}_u$, in two different simulation scenarios. The first one consists in an obstacle free circuit, in which a squared sheet is to be transported to a different location and deformed into a regular hexagonal shape. A formation of $N = 6$ robots grasps the sheet, modelled in 3D with the As-Rigid-As-Possible (ARAP) technique [117]. Only the 2D projection of the model on the horizontal plane is considered by the control algorithm. Then, the control input $\mathbf{U}$ is applied to the robots with the following parameters: $k_{1H} = 4$, $k_{2H} = 2$, $k_{1G} = 4$, $k_{2G} = 2$, $\alpha_H = 1$, $\alpha_G = 10$, $k_{1s} = 3$, $k_{2s} = 1$, $k_{1g} = 0.2$, $k_{2g} = 1$, $k_{1\theta} = 3$, $k_{2\theta} = 1$, $s_d = 1$, $\theta_d = 0$ and the control time step is 0.01 [s]. Figure 5.2 shows the top view of the test case and the evolution of the formation parameters. We can see that there is a smooth transition from the initial to the goal configuration, that is achieved with control inputs and control errors that tend to zero. Note that the robots do not permute their positions in the process, preventing in this manner undesired deformations of the object.

In the next test case we include three obstacles of 1 [m] radius, two static and one dynamic, which must be avoided to reach the goal configuration. Given that the formation controller does not consider explicitly collision avoidance in the formulation, we apply $\mathbf{U}_B$ through the QP-based algorithm. The QP controller is configured with the same control parameters than in the previous case, and the additional ones that follow: $\varepsilon = 3\, h_{ij}$, $\varepsilon' = h'_{ij}$, $\alpha_i = 10$ [m/s^2], $D_{ij}^{min} = 0.5 + d_r$ [m] for the robot-to-robot collision avoidance CBFs, where $d_r = 0.35$ [m] is the diameter of the robots, $D_{ij}^{min} = 1 + d_r$ [m] for the robot-to-obstacle and object-to-obstacle collision avoidance CBFs, and $D_{ij}^{max} = 5$ [m] $\forall i \neq j$. We also include 18 virtual robots, evenly distributed in the formation contour edges. We see in Fig. 5.3 that the goal configuration is successfully reached without collisions, since the minimum distance between the mesh nodes and the nearest obstacle is always above the obstacles radius (1 [m]).

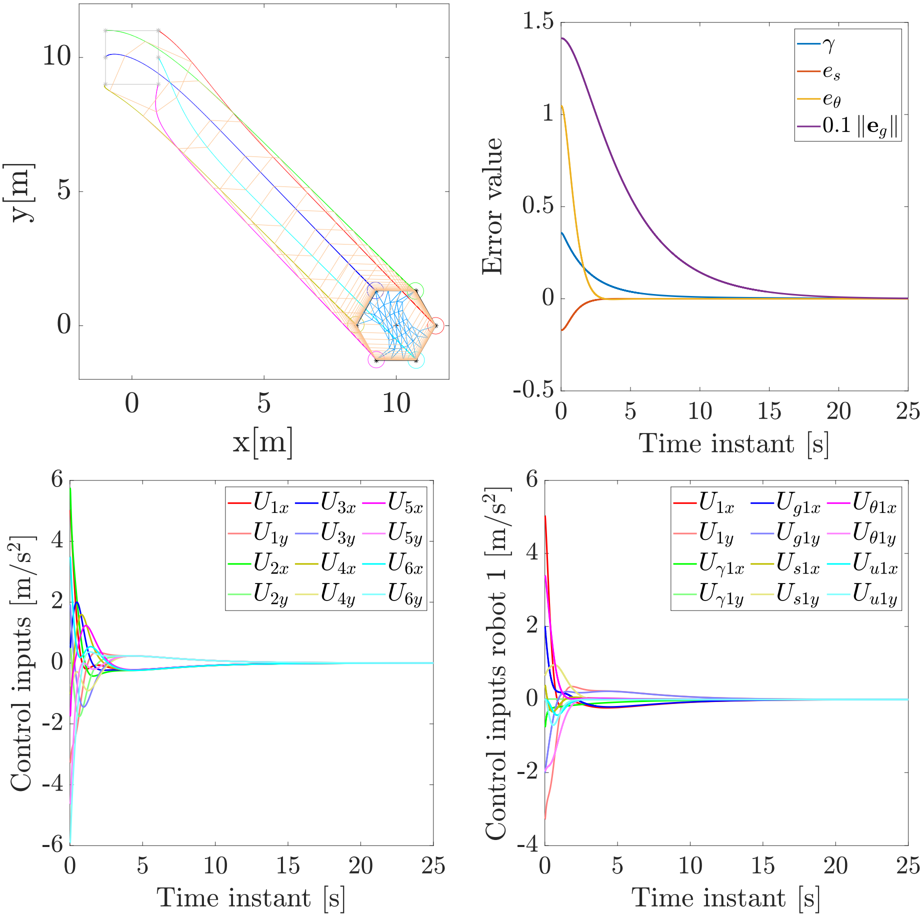


Figure 5.2: From left to right and from top to bottom, top view of the test scenario at $t = 25$ [s], control errors plot, full controller inputs plot and robot 1 control inputs plot (the control inputs plots of the remaining robots are similar). In the first plot, the deformable object (blue mesh) is transported by $N = 6$ robots, from a square to the hexagonal goal formation at the bottom right corner of the plot. It can be seen that the desired shape, scale, rotation and position of the formation are reached with a smooth transition from the initial state, and the control inputs tend to zero over time.

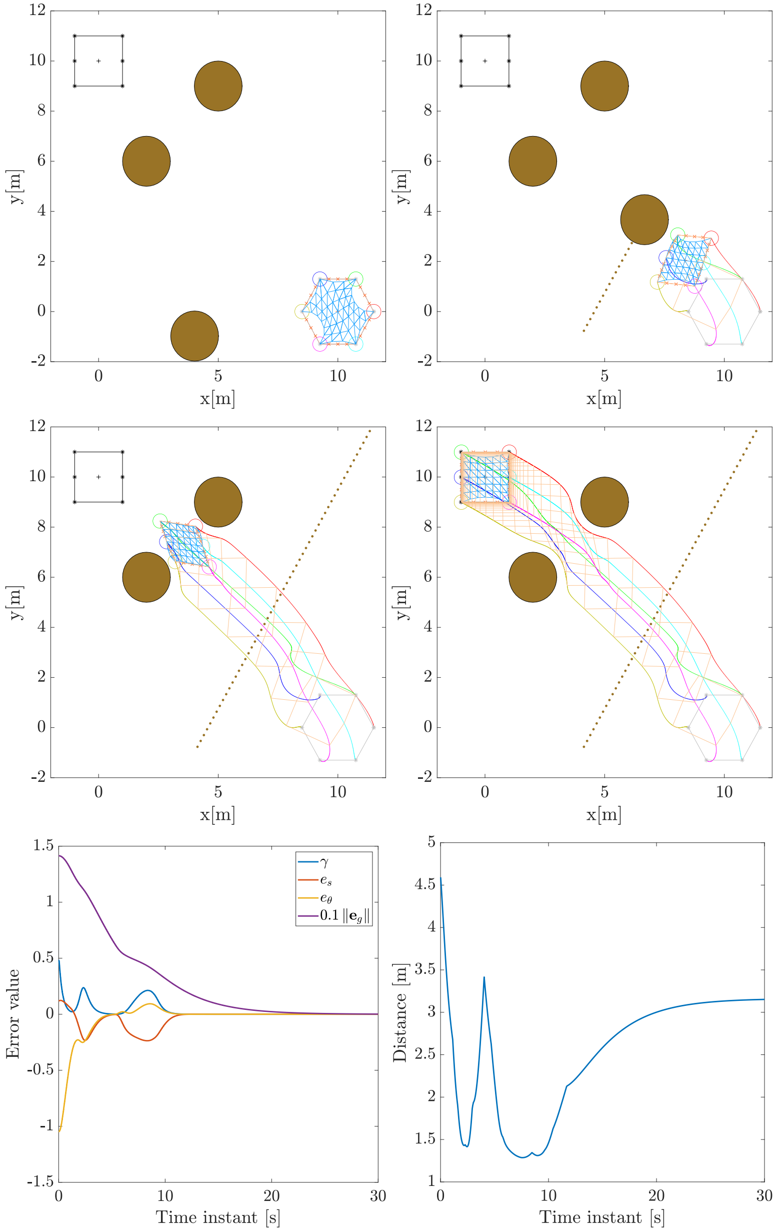


Figure 5.3: From left to right and from top to bottom, four top-view snapshots ($t = 0$ [s], $t = 2$ [s], $t = 8$ [s], $t = 30$ [s]) of the second test scenario with two static and one dynamic obstacles, errors plot and minimum distances plot. This last plot indicates the minimum distance between the mesh nodes and the nearest obstacle. It can be seen that the goal configuration is achieved without collisions between the object and the obstacles (the minimum distance is always above 1 [m]).

5.8.2 Experimental tests

The performance of the proposed controller in real scenarios is tested in two different setups, without and with obstacles. We consider the Robotarium system for these real-world experiments [134]. This system consists in a 3.2 × 2 [m] arena with up to 20 GRITSBot X robots available, which are coordinated by a central unit. A motion capture system provides the position and orientation of the agents at a frequency of 30Hz. Once our system is validated in the Robotarium simulator for Matlab, the software is sent to the real platform through the web application. Is it worth to mention that the robots of this system follow unicycle kinematics, instead of double-integrator ones. Therefore, we integrate the acceleration outputs and then transform the resulting velocities to the unicycle model by means of a diffeomorphism (provided by the Robotarium software).

A 0.6 × 0.75 [m] deformable virtual sheet (real objects are not available for testing in the Robotarium) is transported in the first test case by six robots, one at each corner and the other two in the middle of each one of the longer edges. The goal configuration is a rectangle with $\mathbf{g}_d = [-0.75, 0.15]^\intercal$ [m] (the origin of coordinates is at the center of the arena), $s_d = 1$ and $\theta_d = 0$ [rad]. We configure the controller $\mathbf{U}$ with the following parameters in the overdamped region (see Subsection 5.7.1): $k_{1H} = 0.1$, $k_{2H} = 0.65$, $k_{1G} = 0.1$, $k_{2G} = 0.65$, $\alpha_H = 4$, $\alpha_G = 2$, $k_{1s} = 0.1$, $k_{2s} = 0.3$, $k_{1g} = 0.02$, $k_{2g} = 0.3$, $k_{1\theta} = 0.1$, $k_{2\theta} = 0.3$ and the control time step is 0.033 [s]. Figure 5.4 shows the results of this test. It can be seen that $\mathbf{P}_T$ is reached with near-zero errors, despite the noise and perturbations coming from the real setup (conversion of the control inputs from the double-integrator model to the unicycle one, measurement and actuation errors, etc).

In the second test, the idea is to represent a realistic manipulation scenario in which a rectangular deformable sheet is transported, without collisions with the obstacles present in the environment, to a deformed configuration. A virtual sheet of 0.45 × 0.56 [m] is grasped by four robots at its corners and must be transported to a target configuration. The target configuration is a trapezoid with $\mathbf{g}_d = [-0.8, 0.25]^\intercal$ [m], $s_d = 1$ and $\theta_d = 0$ [rad]. Two obstacles of 0.16 [m] radius, one static and one dynamic, are considered. We configure the controller $\mathbf{U}_B$ with the following parameters: $k_{1H} = 0.1$, $k_{2H} = 0.2$, $k_{1G} = 0.05$, $k_{2G} = 0.2$, $\alpha_H = 4$, $\alpha_G = 2$, $k_{1s} = 0.05$, $k_{2s} = 0.3$, $k_{1g} = 0.01$, $k_{2g} = 0.3$, $k_{1\theta} = 0.05$, $k_{2\theta} = 0.2$ and the control time step is again 0.033 [s]. In turn, the CBFs are configured with: $\varepsilon = 30\,h_{ij}$, $\varepsilon' = h'_{ij}$, $\alpha_i = 20$ [m/s^2], $D_{ij}^{min} = 0.3$ [m], and $D_{ij}^{max} = 5$ [m] $\forall i \neq j$. We include 12 virtual robots, evenly distributed in the formation contour edges. Figure 5.5 shows the results from this test. Note that the task errors converge to a near-zero value and the goal configuration is reached. Note also that the minimum distance between a node of the object's mesh and the center of an obstacle is always above 0.16 [m], which indicates that the object does not collide with the obstacles. We provide additional results in the attached video.

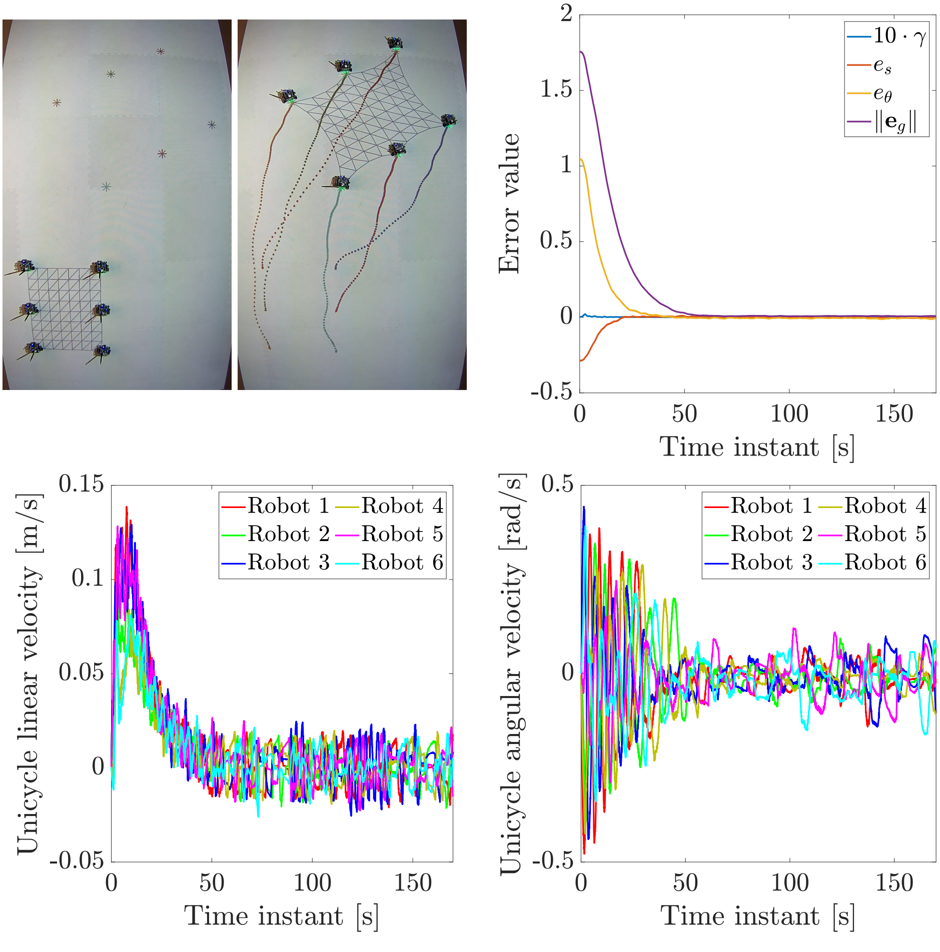


Figure 5.4: From left to right and from top to bottom, top view of the test scenario initially and at $t = 165$ [s], control errors plot, unicycle linear velocities plot and unicycle angular velocities plot. The top-view plots are displayed rotated by 90 [deg]. In the two first plots, a large deformable object is transported and deformed by $N = 6$ robots, from the rectangular initial formation seen in the first plot to a scaled and rotated rectangle seen at the top of the second plot. It can be seen that the desired shape, scale, rotation and position of the formation are reached up to near-zero errors.

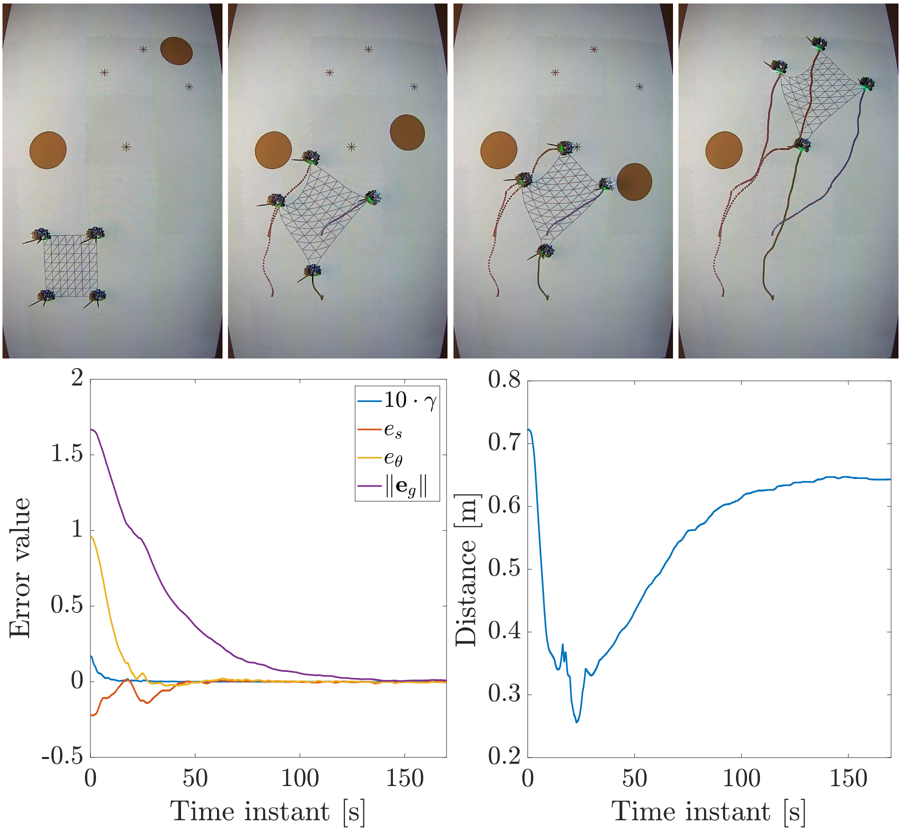


Figure 5.5: From left to right and from top to bottom, four top-view snapshots ($t = 0$ [s], $t = 15$ [s], $t = 22$ [s], $t = 165$ [s]) of the second test scenario in the Robotarium, with one static and one dynamic obstacles, control errors plot and plot of minimum distances between the mesh nodes and the nearest obstacle. The top-view plots are displayed rotated by 90 [deg]. It can be seen that the errors evolve to a stationary near-zero value. The minimum distances are always above 0.16 [m], the radius of the obstacles, which indicates that no collisions occur between the object and the obstacles.

5.9 Discussion

The presented formation controller allows steering a deformable object to a specific configuration in 2D, by means of a team of robots. It includes different terms that modify the shape, scale, position and orientation of the robotic formation to the desired ones. We have demonstrated the existing uncoupling between the problem variables and the stability and convergence of the system. The performance of our proposal has shown successful results in simulated and real scenarios, without and with obstacles.

We have assumed in this chapter that the manipulated object is resistant enough under standard grasping techniques. However, if the object is fragile we should develop alternative manipulation methods for avoiding damages. In the next chapter, we consider thin fragile deformable objects lying on a flat surface. Then, we propose a new manipulation strategy based on sequential planar pushing, which enables object-compliant manipulation of materials with unknown mechanical properties.

Chapter 6

Multirobot pushing of thin fragile deformable objects

For the goal of driving a thin fragile deformable object to a target position and orientation, we propose in this chapter a manipulation method based on executing planar pushing actions on the object edges with a team of robotic arms. Firstly, we obtain a probabilistic model through Gaussian process regression, which represents the time-varying deformation properties of the system. Then, we exploit the model in the framework of an Adaptive Bayesian Optimization (ABO) algorithm to compute the optimal pushing action at each instant. The exploration of the solution space is accelerated by means of a novel set of constraint adaptation techniques, designed to find effective pushing actions. We illustrate the performance and validity of our proposal in different simulation scenarios.

6.1 Introduction

Objects that can easily be damaged when manipulated by robotic platforms, due to their fragile structure or the manipulation strategy, are common in certain areas. Examples of deformable fragile objects include fish and meat portions, food dough, unfired ceramic pieces and plastic parts at high temperatures. The field of manipulation of fragile deformable objects has gained attention in the last years, with solutions for grasping delicate objects by exploiting the environmental constraints [123] and for cleaning deformable parts with fragile sections [83], among others. For such fragile materials, the control actions must be limited and the manipulation strategy must be carefully designed, so that their integrity is not compromised in the process. In this chapter, we propose a new method to manipulate thin fragile deformable objects based on a sequence of planar pushing actions. These actions aim at driving the object to a specific configuration (position in 2D and orientation around the vertical axis), while preserving its shape within an admissible range of deformation. In contrast to the standard prehensile manipulation of rigid or semi-rigid objects, pushing is less invasive and, therefore, more suitable for manipulating thin fragile objects that can be damaged when

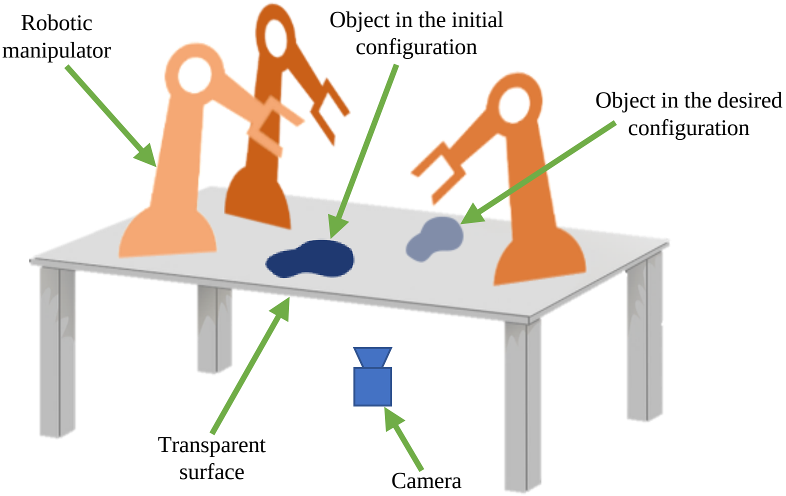

Robotic manipulator
Object in the initial configuration
Object in the desired configuration
Transparent surface
Camera

Figure 6.1: System overview: A fragile thin deformable object is pushed by multiple robotic manipulators to a desired configuration through object-compliant actions that preserve its integrity. The input information (object contour) to the Adaptive Bayesian Optimization-based method is provided by a camera under the transparent surface that supports the object.

grasped with robotic tools. It is also advantageous in this context to consider multiple manipulators, for avoiding reachability issues.

Manipulation of rigid objects by pushing is a mature topic in robotics [121]. A key work is based on Variational Heteroscedastic Gaussian processes (the amount of noise introduced by the system depends on the action) to model planar pushing interaction [19]. An uncalibrated image-based controller is considered in another study for parking objects by pushing [92]. The problem of pushing objects with unknown center of mass is tackled in a recent work by finding a suitable two-edge-contact configuration [50]. However, the number of solutions for robotic non-prehensile manipulation of deformable objects is still limited. Some works in this context include the results of the RoDyMan project, focused on dynamic non-prehensile manipulation of rigid and deformable objects with robotic platforms [110]. Modeling the deformation of an object under pushing has been addressed by offline learning with integration of visual and force information [12]. Different studies tackle non-fixed contact manipulation configurations by means of optimization-based methods for contact adjustment [70] and contact point selection [53]. Recently, a strategy for object rearrangement based on planar pushing has been analyzed and validated [26]. In comparison with our proposal, these works only consider partially and not fully deformable objects.

Our proposed method is based on a Bayesian Optimization algorithm. Bayesian-Optimization methods are focused on finding the optimum value of an unknown function, in a small number of iterations. In each iteration of the algorithm, a probabilistic model (typically a Gaussian Process) is updated through Bayesian inference, with data selected by means of an acquisition function. A pioneer work solves the problem of robotic motion planning in environments with deformable objects by means of Gaussian Process Regression [49]. Safe and robust robotic grasping is tackled in another study by haptic exploration and unscented Bayesian optimization [103]. With respect to the manipulation of deformable objects, a solution has been proposed based on servo control and a deformation model learned with fast online Gaussian process regression [69]. Another study considers Bayesian inference and deep learning for solving real-to-sim problems of deformable objects manipulation [8]. Other applications of Bayesian optimization include non-rigid structure from motion [1] and soft landing control of reluctance actuators [97].

Our proposal contributes in different ways to the current state of the art. For the first time, we apply the technique of sequential pushing to fully deformable objects, with pushing actions that are computed by means of an Adaptive Bayesian Optimization-based algorithm. This algorithm takes into account the temporal evolution of the object, that deforms after every pushing action, and learns a policy to drive the object to a planar goal configuration. In addition, we propose a novel set of techniques for adapting the problem constraints that improve the model learning process, in terms of speed and accuracy.

6.2 Manipulation task overview

The manipulation setup is equipped with multiple robotic arms and consists of a flat horizontal surface in which a thin fragile deformable object lies. A camera is placed in the environment so that the 2D contour of the object is detected without occlusions. In this case, the object contour corresponds to the projection in the horizontal surface along the vertical axis, and it is defined by the ordered sequence of contour points $\mathbf{V}_c = [\mathbf{v}_{c1}, \mathbf{v}_{c2}, ..., \mathbf{v}_{cM}] \in \mathbb{R}^{2\times M}$. We consider that the state of the object is determined by the contour centroid $\mathbf{c} \in \mathbb{R}^2$, the orientation of the contour around the vertical axis $\theta \in [0, 2\pi)$ and the shape of the contour represented as a binary mask $\boldsymbol{\gamma}$.

The goal is to perform object-compliant actions that transport and rotate the object to a goal position. Due to the thin and fragile structure of the object, it cannot be grasped and lifted, and its deformation must be bounded along the process. We define three different errors to assess the task completion:

$$e_c = \|(\mathbf{c} - \mathbf{c}_d)\| / \|\mathbf{c}_d\| \ , \tag{6.1}$$

$$e_\theta = (\theta - \theta_d)/\pi \ , \tag{6.2}$$

$$e_\gamma = \Sigma(\boldsymbol{\gamma} \oplus \boldsymbol{\gamma}_d)/\Sigma\, \boldsymbol{\gamma}_d \ , \tag{6.3}$$

where $\|\cdot\|$ is the 2-norm operator and the parameters with d subscript refer to the user-defined target values. The contour centroid is computed as $\mathbf{c} = \frac{1}{M}\mathbf{V}_c\mathbf{1}_M$, with $\mathbf{1}_M$ being a column vector of M ones. We apply Principal Component Analysis (PCA) to the set $\mathbf{V}_b =$

$\mathbf{V}_c - \mathbf{c}\,\mathbf{1}_M^\intercal = \mathbf{USV}^\intercal \in \mathbb{R}^{2\times M}$, and we obtain the orientation of the contour as $\theta = \mathrm{atan}(V_{2,1}, V_{1,1})$, where the subscripts refer to the corresponding elements of $\mathbf{V}$.

Equation (6.3) represents a ratio, where the numerator corresponds to the difference in pixels between the current $\boldsymbol{\gamma}$ and the desired $\boldsymbol{\gamma}_d$ shapes of the object. $\boldsymbol{\gamma}$ is the binary mask of $\mathbf{R}(-\theta)\,\mathbf{V}_b$, where $\mathbf{R}(-\theta)$ is the rotation matrix of $-\theta$ radians, and $\boldsymbol{\gamma}_d$ is the binary mask of the desired contour with centroid at the origin and whose principal direction is aligned with the x axis. The difference is obtained as the summation of the non-zero pixels resulting from the boolean XOR operation ($\oplus$ symbol) between $\boldsymbol{\gamma}$ and $\boldsymbol{\gamma}_d$. In turn, the denominator is the sum of non-zero pixels of the desired shape ($\boldsymbol{\gamma}_d$). Note that γ_d is not necessarily an exact shape we seek, but a shape in which the deformation of the object is bounded.

Then, the total task error is

$$e_T = k_c\, e_c + k_\theta\, |e_\theta| + k_\gamma\, e_\gamma \ , \tag{6.4}$$

where k_c, k_θ and k_γ are positive weights and $|\cdot|$ is the absolute value operator. Note that $e_T \geq 0$.

The method we propose for completing the manipulation task consists in executing a sequence of pushing actions $\mathbf{u} = (\kappa, d)$ via the end-effector of the arms. These pushing actions aim at reducing the task error e_T over time, while preserving the integrity of the object. Two distinct parameters determine the actions: $\kappa \in [0, 1]$, defined as the object contour ratio, and $d \in [d_{min}, d_{max}]$, which stands for the pushing distance. The contour ratio is a fraction of the effective contour $\mathbf{V}_e \subset \mathbf{V}_s$, where $\mathbf{V}_s$ is the augmented contour with

$$\mathbf{v}_{si} = \mathbf{v}_i + d_o\, \mathbf{v}_{bi}/\|\mathbf{v}_{bi}\| \ , \tag{6.5}$$

where $\mathbf{v}_{bi} = \mathbf{v}_i - \mathbf{c}$ and d_o is a safety offset to take into account the width of the pushing tool. Therefore, κ represents the position of the effective contour between the first effective point $\mathrm{v}_{e1}(\kappa = 0)$ and the last point $\mathrm{v}_{eM_e}(\kappa = 1)$, where M_e is the number of points of the effective contour. d is always applied perpendicularly to the contour segment. Note that the robotic arm that will perform the pushing action at every time instant will be the one closest to the pushing point.

6.3 Optimization problem

We can express the manipulation task at the time instant k as the following optimization problem:

$$\begin{aligned} \text{Given} \quad & \mathbf{V}_c,\ \mathbf{c}_d,\ \theta_d,\ \gamma_d,\ f(\mathbf{x}) \\ \underset{\mathbf{u}}{\text{minimize}} \quad & \xi = e_{T(k+1)}(f(\mathbf{x})) - e_{Tk} \\ \text{subject to:} \quad & \kappa_{min} \leq \kappa \leq \kappa_{max} \ , \\ & d_{min} \leq d \leq d_{max} \ , \end{aligned} \tag{6.6}$$

where $\mathbf{x} = (\mathbf{u}, t) = (\kappa, d, t) \in \mathbb{R}^3$, $f(\mathbf{x})$ is an unknown function that maps κ and d with e_T, over time. Therefore, we want to find at every instant k the action that most decreases the error with respect to the previous instant and satisfies the existing constraints.

Different physical effects are integrated in f. The ones with greater impact on the development of the task are the deformation modes of the object and the friction between the object and the manipulation surface. In general, fragile deformable objects may show deformations that disappear when the acting force is released (elastic) or deformations that remain afterwards (plastic). Besides, the manipulation actions may create compression, traction, shear or torsion among other deformation modes. With respect to friction, we can differentiate between static friction, which happens when the object is not globally in motion, and dynamic friction, which acts during global motion. The transition between these modes determines the difference between actions that induce local motions only (deformations) and the actions that produce both local and global motions (deformation and displacement).

These effects and their interactions are complex to model, and their variability is high. This is the reason why obtaining f for each manipulation case seems appropriate. The method we propose is based on the Adaptive Bayesian Optimization (ABO) technique [105]. While a standard Bayesian Optimization (BO) approach allows obtaining the global optimum of an unknown time-invariant function, ABO allows to track the minimum of a time-varying function over time. This method can be divided in three main steps:

1. *Constraints adaption.* In order to improve the performance of the system, in terms of accuracy and speed, the optimization constraints are adapted according to the state of the object.

2. *Data acquisition.* At the beginning of the process, the parameter space (delimited by the constraints) is randomly sampled for obtaining an initial model. Then, the acquisition process modulates to sampling points that raise the highest probability of improvement.

3. *Model update.* The model is updated with new data after every sampling. This allows refining the knowledge about the underlying physics and the evolution of the system over time.

In the present case, sampling requires pushing and detecting the contour afterwards. Pushing modifies not only the state of the object, but also its deformation and friction properties, which may vary due to local stiffening, creasing and other effects.

6.4 Gaussian process regression

The method we propose requires a prior regression model. Under the assumption that the variability of the unknown model of our system, in space and time, can be predicted by a Gaussian probability function, we consider a spatiotemporal Gaussian Process (GP) as the required surrogate model. GPs are likely the most popular regression models for BO, and they represent the generalization of the Gaussian probability distribution [108]. This model is specified by a mean function $\mu_0(\mathbf{x})$ and a covariance function, or kernel, $k(\mathbf{x}, \mathbf{x}')$ as $f(\mathbf{x}) \sim \mathcal{GP}(\mu_0, k)$. We assume that $\mu_0 = 0$ for convenience, and we consider the kernel to be stationary, separable and of the form

$$k(\mathbf{x}, \mathbf{x}') = k_s(\mathbf{u}, \mathbf{u}') \cdot k_t(t, t') \ , \tag{6.7}$$

where k_s and k_t are the spatial and temporal kernel functions, respectively [105]. We consider kernel functions of the Matérn 52 form:

$$\begin{aligned} k_{M52}(\mathbf{x}, \mathbf{x}') = \sum_i \Bigg(\sigma_f^2 + \frac{\sqrt{5}\|x_i - x_i'\|_2}{\rho_i} \\ + \frac{5\|x_i - x_i'\|_2^2}{3\rho_i^2} \exp\Bigg(-\frac{\sqrt{5}\|x_i - x_i'\|_2}{\rho_i} \Bigg) \Bigg) \ , \end{aligned} \tag{6.8}$$

where σ_f^2 is the characteristic variance and ρ_i is the length scale of every dimension (with $i \equiv \kappa$, d, t).

For updating the model with the sets of S measurements $\mathbf{Y} = [\xi_1, \xi_2, ..., \xi_S] \in \mathbb{R}^{1\times S}$ and $\mathbf{X} = [\mathbf{x}_1^\intercal, \mathbf{x}_2^\intercal, ..., \mathbf{x}_S^\intercal] \in \mathbb{R}^{3\times S}$ we must compute the posterior Gaussian distribution

$$f|\mathbf{X}, \mathbf{Y}, \mathbf{x} = \mathcal{N}(\mu, \sigma) \ , \tag{6.9}$$

where μ and the variance σ^2 are computed with the Sherman-Morrison-Woodbury formula as

$$\mu = \mathbf{k}^\intercal \mathbf{K}^{-1} \mathbf{Y} \ , \tag{6.10}$$

$$\sigma^2 = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}\mathbf{K}^{-1}\mathbf{k} \ . \tag{6.11}$$

$\mathbf{K} \in \mathbb{R}^{S\times S}$ is the covariance matrix with $\mathbf{K}_{ij} = k(\mathbf{X}_i, \mathbf{X}_j)$ and $\mathbf{k}$ is defined with $\mathbf{k}_i = k(\mathbf{X}_i, \mathbf{x}), \forall i, j \leq S$.

As mentioned before, we want to track the minimum of ξ over time. For efficiently getting a good estimate of this value, the variability of the solution search space must be represented well enough in the initial model. We apply the Latin Hypercube Sampling (LHS) technique, that allows covering the search domain in a near-random manner with S_0 initial samples. Algorithm 6.3 reports the main steps involved in getting the initial model.

6.5 Constraints adaption

Learning to compensate three different kinds of errors (position, orientation and shape) at the same time with a single manipulator is a challenging task, even for BO-based techniques.

Algorithm 6.3 Get initial model $f(\mathbf{x})$

```
Require: $S_0, \mathbf{V}_c, \mathbf{c}_d, \boldsymbol{\theta}_d, \boldsymbol{\gamma}_d$
$\mathbf{U} \leftarrow \text{LHS}(S_0)$ # set of random pushing parameters
$S \leftarrow 0$
while $S < S_0$ do
    $\mathbf{u} \leftarrow \mathbf{U}(S)$
    $\mathbf{V}_c, \mathbf{X}, \mathbf{Y} \leftarrow applyPushingAction(\mathbf{u})$
    $S \leftarrow S + 1$
end while
return $f(\mathbf{x}) \leftarrow updateModel(\mathbf{X}, \mathbf{Y}), \mathbf{V}_c$
```

This is due to the fact that, at some point, decreasing one of the errors may imply that another one is increased. E.g., when rotating the object for correcting its orientation, the object could deform increasing the shape error. Besides, despite the fact that exploring the solution domain is a necessary step in ABO, we want to minimize the number of pushing actions that, for the sake of exploration, increase e_T.

Therefore, to avoid learning inefficient policies that can lead to undesired loops, we apply the criterion of delimiting the search to those parameter subspaces that are likely going to yield appropriate responses. Note that due to the dynamic nature of our problem, it is required to modify the constraints according to the state of the object at every time step. We describe next the different adaption strategies that we apply to the constraints of our system.

6.5.1 Effective contour

The pushing direction should be as close as possible to the error direction in order to minimize the cost of pushing an object to a target location. Based on this fact, we propose to obtain the effective contour $\mathbf{v}_e$, i.e. the fraction of the contour we consider for pushing, according to the following logistic function:

$$a = \frac{a_{max} - a_{min}}{1 + \exp\Big(r_a\,(e_c - e_{cm})\Big)} + a_{min}\ , \tag{6.12}$$

where a_{max} is the function supremum, r_a is the growth rate, e_{cm} is the midpoint of the function and a_{min} is the offset. a represents the angle of a circular sector that covers the region of $\mathbf{v}_s$ from which we want to extract $\mathbf{v}_e$. This kind of logistic function is convenient because it provides a soft transition between the minimum and maximum angles a_{min} and a_{max}, respectively. As it can be seen in Fig. 6.2, the total angle is computed as $2\pi a$, is measured around $\mathbf{c}$ and is divided in half in the direction of $-(\mathbf{c} - \mathbf{c}_d)/\|\mathbf{c}_d\|$. The purpose of this constraint is to limit the search of pushing points to those that are likely going to reduce e_c. At first the angle is sharp, and as e_c decreases it opens more and more, until most of it is

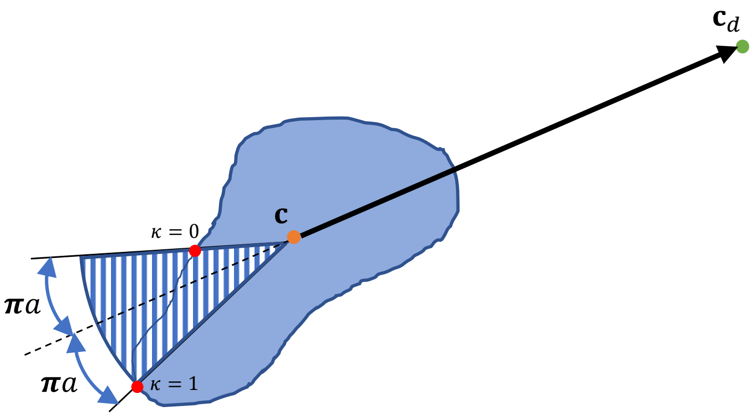


Figure 6.2: The effective object contour is computed as a fraction of the circular sector of $2\pi a$ radians. As depicted, this section is placed around $\mathbf{c}$ with a total angle of $2\pi a$ radians, and is divided in half by the direction $-(\mathbf{c} - \mathbf{c}_d)/\|\mathbf{c}_d\|$.

considered when the error is close to zero. This allows considering more tentative points for decreasing e_θ and e_γ.

6.5.2 Maximum pushing

Coming back to the notions of static and dynamic friction, it can be inferred that for globally translating the object we must counteract the resistance to motion static friction exerts. This can be done with long pushes, that deform but also translate the object. However, for the shape and orientation correction, short pushes that do not produce global motion are generally sufficient. In order to guide the learning process into a coarse/fine mode shift, we can modify the maximum pushing length d_{max} according to e_c as follows:

$$b = \frac{1 - b_{min}}{1 + \exp\Big(- r_b \left(e_c - e_{cm}\right)\Big)} + b_{min} \ . \tag{6.13}$$

We can see that this function is designed similarly to (6.12). Consequently, 1 and r_b are the function supremum and the growth rate, respectively. In this case, we consider $d_{max} = b \cdot \hat{d}_{max}$, where $\hat{d}_{max}$ is the absolute maximum. Note that b evolves from 1, when $e_c - e_{cm}$ is high, to a minimum b_{min}, when the error is close to zero. This ensures that the absolute maximum pushing length is considered in the search space at the beginning, and it decreases as e_c does in a transition towards a fine-mode manipulation.

6.5.3 Rotation error compensation

The last constraint adaption technique that we propose aims at improving the orientation correction learning. First, we define the function $\lambda(\mathbf{w}_1, \mathbf{w}_2)$ for two generic vectors $\mathbf{w}_1$ and $\mathbf{w}_2$ as

$$\lambda(\mathbf{w}_1, \mathbf{w}_2) = \star(\mathbf{w}_1 \wedge \mathbf{w}_2) \ , \tag{6.14}$$

where '$\wedge$' denotes the 2D wedge product and '$\star$' is the Hodge star operator. This function gives us the direction of the angular error around the vertical axis. Then, we propose to obtain κ_{min} and κ_{max} from the longer section of the effective contour in which the following identity holds $\forall i$:

$$\operatorname{sign}(\theta - \theta_d) = \operatorname{sign}\Big(\lambda(\mathbf{n}_i, (\mathbf{v}_{e(i+1)} + \mathbf{v}_{ei})/2 - \mathbf{c})\Big) \ , \tag{6.15}$$

where $\mathbf{n}_i$ is the normal vector of segment $\mathbf{s}_i = [\mathbf{v}_{ei}, \mathbf{v}_{e(i+1)}]$, pointing outside the contour. This identity indicates that a pushing action applied perpendicularly to segment $\mathbf{s}_i$ is likely going to rotate the object so that the rotation error decreases.

While this constraint adaption maximizes the probability of correcting the rotation error at every pushing attempt, it is also more restrictive than the previous ones. Depending on the deformation state of the object and the length of the effective contour, the considered section $[\kappa_{min}, \kappa_{max}]$ might be too short to include good candidates for correcting the position error. This is why we set the condition $e_c < e_{cf}$ to decide when to apply this constraint adaption, where e_{cf} is a threshold.

6.6 Acquisition function

One of the key aspects of ABO is that it allows tracking the minimum of $f(\mathbf{x})$ through successive trials or acquisitions where the value the search criterion yields is maximum. The acquisition step, in our case, should consider a balance between exploration and exploitation. Once the algorithm finds a suitable pushing action, this action should be exploited until its effectiveness decreases. In that moment, the algorithm should explore the solution search space to find a better pushing action candidate.

The dynamic acquisition function we consider is based on the Lower Confidence Bound (LCB) [29]:

$$\mathrm{LCB}(\mathbf{x}) = \mu(\mathbf{x}) - w\sigma(\mathbf{x}) \ , \tag{6.16}$$

where $w \geq 0$ is a constant that regulates the confidence level. According to this function, the lower the LCB value the more promising a sample is. We look for a minimum LCB by optimizing the function under the problem constraints κ_{min}, κ_{max}, d_{min} and d_{max}. The Adam optimization algorithm provides a suitable solution by means of exponential moving averages of the gradient and the squared gradient of the function. At every iteration of the method the search space boundaries are checked, and the evaluated point is saturated to the limits in case the constraints are violated. After a user-defined number of iterations, the best solution is selected as the candidate pushing action for the next time step.

It is worth mentioning that, in contrast to other applications of ABO, we do not optimize the time step in which we sample/push the system. This is due to the fact that the temporal evolution of the system is conditioned to the pushing step. Then, we are interested in applying the pushing actions as soon as possible, so that the operation time is minimized. We show the complete structure of our pushing method in Algorithm 6.4.

Algorithm 6.4 Find optimal pushing action $\mathbf{u}$ at time t.

```
Require: $\mathbf{X}, \mathbf{Y}, \mathbf{V}_c, \mathbf{c}_d, \boldsymbol{\theta}_d, \boldsymbol{\gamma}_d, \mu, \sigma$
$\mathbf{c} \leftarrow 1/M\, \mathbf{V}_c \mathbf{1}_M$
$\boldsymbol{\theta} \leftarrow \mathrm{PCA}(\mathbf{V}_c)$
$\boldsymbol{\gamma} \leftarrow binaryMask(\mathbf{V}_c)$
$e_c \leftarrow \|(\mathbf{c} - \mathbf{c}_d)\| / \|\mathbf{c}_d\|$
$e_\theta \leftarrow (\theta - \theta_d)/\pi$
$e_\gamma \leftarrow \Sigma(\boldsymbol{\gamma} \oplus \boldsymbol{\gamma}_d)/\Sigma\, \boldsymbol{\gamma}_d$
while $(e_c \geq e_c^{stop})$ & $(e_\theta \geq e_\theta^{stop})$ & $(e_\gamma \geq e_\gamma^{stop})$ do
    $\kappa_{min}, \kappa_{max}, d_{min}, d_{max} \leftarrow constAdaption(e_T)$
    $\mathbf{u} \leftarrow acquisition(\mu, \sigma)$
    $\mathbf{V}_c, \mathbf{X}, \mathbf{Y} \leftarrow applyPushingAction(\mathbf{u})$
    $f(\mathbf{x}) \leftarrow updateModel(\mathbf{X}, \mathbf{Y})$
    $\mathbf{c} \leftarrow 1/M\, \mathbf{V}_c \mathbf{1}_M$
    $\boldsymbol{\theta} \leftarrow \mathrm{PCA}(\mathbf{V}_c)$
    $\boldsymbol{\gamma} \leftarrow binaryMask(\mathbf{V}_c)$
    $e_T \leftarrow computeTotalError(\mathbf{c}, \boldsymbol{\theta}, \boldsymbol{\gamma})$
end while
return $e_T, \mathbf{X}, \mathbf{Y}, \mathbf{V}_c$
```

6.7 Simulation results

We evaluate our method by means of two different simulation scenarios in the Pybullet environment, a module in Python language focused on sim-to-real transfer [43]. Note that, for simplicity, only one arm is considered in these tests, but the method we propose is valid for other configurations with multiple manipulators.

In the first test, we consider a $0.2 \times 0.1 \times 0.02$ [m] (length × width × depth) flat rectangular deformable object, modelled with the finite element method (FEM). The object is placed over a transparent table, and it is detected by a virtual camera beneath the table. We extract the contour of the object by means of the OpenCV library, by color thresholding in the HSV space. The manipulator model is a Kuka iiwa arm attached to the table at a certain distance

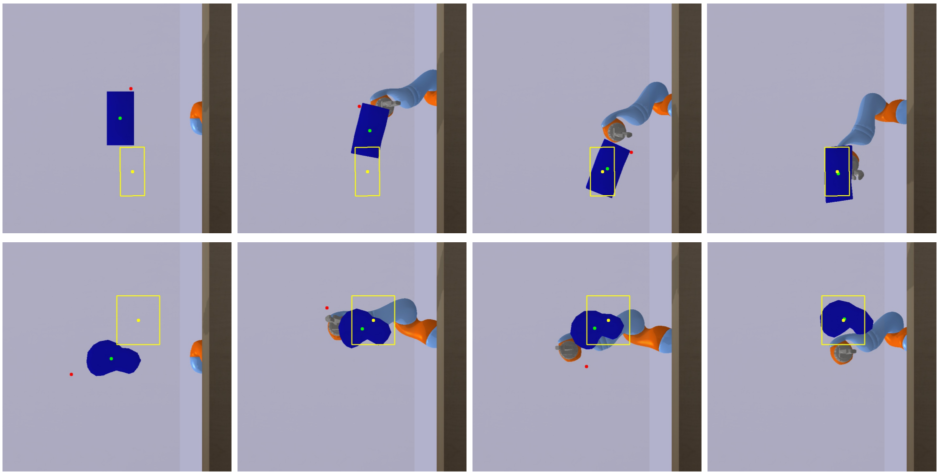


Figure 6.3: Sequence of four bottom-view snapshots for two scenarios, a rectangular deformable object (top row) and a non-regular shaped deformable object (bottom row). The snapshots are acquired at $k = \{0, 10, 18, 53\}$ and $k = \{0, 10, 26, 39\}$, respectively. The deformable object, in blue, is pushed towards the target rectangle, in yellow. Note that the red dot, which indicates the next pushing point, is separated from the contour to compensate the gripper width.

from the object. The goal of the task is to push the object to a target position while preserving its initial shape and orientation. For obtaining the initial model, we apply the space filling technique LHS [98], with $S_0 = 10$ initial samples. Figure 6.3 (top) shows four snapshots of the first experiment, at the initial time and after the 10^{th}, 18^{th} and 53^{th} pushing actions. Figure 6.4 (top) depicts the evolution of the task errors (left), and the evolution of the pushing parameters and the cost function along the task (right). In the latter, the left axis corresponds to the values of the pushing parameters κ and d, while the right axis indicates the values of ξ . Note that after the first 10 pushes, an initial model is learnt, and the position error decreases fast. As we have set the threshold $e_{cf} = 0.15$, the adaptive rotation error compensation is active after $k = 18$. We can see its impact in the errors trend: while the rotation error decreases, the other two start to increase. However, once the model learns to push under the new constraints, the errors start to decrease until they reach the stop values. With respect to the pushing actions, we can see that the system exploits those pushing actions that reduce e_T with respect to the previous instant until they stop being effective. Then, the system iterates to find the pushing action that is most likely going to reduce the error at the current state of the system.

In the second scenario, we place a non-regular shaped object of $0.13 \times 0.2 \times 0.01$ [m] over the table, with different deformation properties than the previous one. The task consists in pushing the object to a target place, by maintaining a compact configuration and preserving the initial orientation. For maintaining a compact configuration, we define a rectangular goal

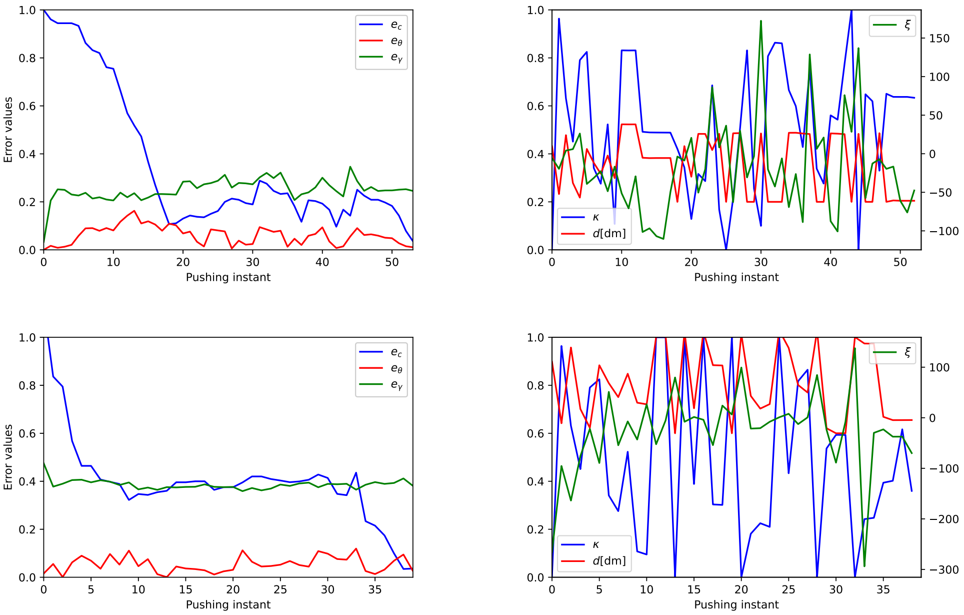


Figure 6.4: Errors and actions plots for the first test scenario (top row) and second test scenario (bottom row). In the first example, e_c decreases fast until the adaptive rotation error compensation activates. After that, the system learns to push under the new constraints, and eventually the errors decrease to the final values. Note how the pushing parameters (left vertical axis) vary when e_T increases with respect to the previous instant, until the pushing actions start to decrease ξ (right vertical axis) again. In the second example, e_c decreases fast at the beginning. After some time, when the system learns to push under the new constraints, the errors finally decrease to the final values. Note that in this case the system is forced to explore more than in the previous one. When the pushing actions stop being effective and ξ increases, the pushing parameters change until ξ starts to decrease again.

shape with aspect ratio close to 1. Note that γ_d can be seen as a deformation constraint for the algorithm. We set $S_0 = 10$, and due to the fact that longer pushes are required to move this object, we increase d_{min} and d_{max} with respect to the previous case. Figure 6.3(bottom) shows the bottom view of the system at four different pushing instants. Figure 6.4(bottom) depicts the evolution of the task errors, the pushing actions and the cost function. In this example, the error decreases fast at the beginning thanks to the effective contour constraint adaption. There is a second fast improvement that stops when e_c reaches threshold e_{cf} and the rotation error compensation activates. After that, the errors continue to decrease with some small oscillations, and finally they reach the stop values. As for the pushing parameters and the cost function values, the same behavior than in the first scenario is shown: the system exploits the actions that decrease ξ until they stop being effective, and then it explores to find new parameters. We include a video as additional multimedia material with the complete pushing sequence of the second experiment.

6.8 Discussion

Manipulating deformable objects is a challenging task for robotic systems due to the large variety of deformations that can appear in the object, which demand specific techniques highly dependent on the material properties. Unlike other kinds of deformable objects, any force applied to a fragile deformable object may alter its shape to the extent that the structural integrity is compromised. Despite the challenges, it is a crucial skill required for robots to manipulate this materials in household and industrial environments. In the present chapter, an algorithm based on Bayesian optimization has been proposed to achieve robotic pushing of thin fragile deformable objects. The proposed approach has multiple advantages. First, the temporal evolution of the object is considered and a policy is learnt to perform a series of pushes on the object to move it from one location to another in a plane. By doing so, the shape of the deformable object is also actively controlled in order to maintain it within an admissible range. Second, in comparison to modeling techniques such as finite element methods, where a precise knowledge of the material property is required, our approach relies on a probabilistic model that is computed in few trial iterations and updated over time to make better predictions. Similarly, in comparison to approaches based on deep learning, the proposed probabilistic model is generic and works well for unknown deformable objects without the need for huge new training data. Third, a novel set of rules to speed up the probabilistic model learning process and its performance are proposed. The approach is successfully validated in simulation with objects of different shapes and properties.

After having dealt with deformable objects of different kinds in a variety of manipulation tasks, we address the final conclusions in Chapter 7. In addition, we summarize the tentative future lines of work we have identified along the previous research topics.

Chapter 7

Conclusions

For the goal of manipulating deformable objects with multiagent systems, we have proposed and validated in this thesis a series of methods that focus on particular tasks. After a comprehensive study of the state of the art, we have covered the perception, transport and deformation control problems of general and more specific manipulation scenarios. We report the main conclusions and tentative future research lines derived from these works in this last chapter.

7.1 Summary and conclusions

Numerous aspects related to the topic of multiagent manipulation of deformable objects have been explored in this thesis. This kind of objects has unquestionable impact in industrial, medical and domestic domains. Through the different chapters, we have shown the interest and potential benefits of automatizing tasks in these areas. Starting with perception, we have described how to perform coverage of deformable objects over time, with minimal multicamera systems. After analyzing the visibility properties of general shapes, we have studied this task in an optimization framework. Then, we have learned about the different optimal results we achieve in global and local solution spaces. It is worth mentioning that our formulation is flexible, and can be applied to different kinds of sensors and configurations with spatial constraints without significant modifications.

After this, and following the natural sequence perception-action, we have investigated the problem of transporting deformable objects with multiple robotic manipulators in obstacle-populated environments. We have explored two different approaches, with and without deformation model. As one of the main contributions of this thesis, we have proposed and validated a model based on the deformable bounding box of the transported object. By reaching a balanced compromise between complexity and accuracy of the represented physics, we have achieved a versatile model that is well suited for real-time transport applications. The alternative method also represents one of the most relevant contributions of this work. It consists in a formation controller that enables the system to reach a target configuration in terms of shape, scale, position and orientation of the robotic formation that grasps the object. We have formally proven the existing uncoupling between the formation variables under the proposed controller, as well as the stability and convergence of the system. In both cases, we have augmented the nominal controllers with control barrier functions, for guaranteeing safety requirements. They have proven to be a powerful and minimally-invasive tool for keeping the system's state within the safe region of performance at all times.

Finally, the last research work of this thesis aims at advancing in the solution to the problem of manipulating thin fragile deformable objects. These objects, not considered in the previous developments, cannot be grasped with standard methods. Instead, we assume they lie on a flat surface, and we propose a new manipulation strategy based on a sequence of planar pushing actions exerted by multiple robotic arms. With this technique we are able to translate and rotate the object, while limiting the applied deformation, until it reaches a target position and orientation in the surface. By means of Gaussian process regression and Adaptive Bayesian Optimization, we obtain a probabilistic model that we utilize to find the most effective pushing action at every time instant. A set of adaptive constraints accelerates the process by restricting the solution search space to the region where the best solutions are more likely going to be found.

7.2 Future research lines

Despite the advancements acquired in this thesis, there is still room for improvement. In this section, we briefly present some tentative lines of research that we think are promising for contributing to the general problem of manipulating deformable objects with multiagent systems.

- Measuring the state of deformable objects under manipulation, with multiple sensors, is a research field with numerous interesting alternatives: exploring techniques that combine different kinds of information; applying redundancy by measuring the same properties with different sensors; and developing sensing techniques that adapt their configuration according to the state of the sensed object or the environmental conditions, among others. Note that all these ideas have the common goal of improving the information that is obtained by either increasing the quantity or the quality of the data.
- Modelling is another dimension of the problem with possible extensions that include: applying new modeling techniques for real-time predictions; obtaining models from different data types; including statistical methods for quantifying uncertainty of the modelled physics; developing models robust to measurement noise; extending generality through versatile formulations; and exploring approaches that integrate modelling and control (e.g. deep learning-based strategies). I. e., the objective is to improve the accuracy, robustness and efficiency of the models.
- Controlling manipulation systems that include deformable objects is a field with potential improvements that may come from different sources: designing controllers for

manipulators with more complex dynamic models; formulating robust controllers under measurement noise and disturbances; exploring control techniques with adaptive capabilities in time-varying scenarios; applying mathematical analysis tools for guaranteeing stability and convergence of the proposed methods, etc. Similarly to the case of modelling, accuracy, robustness, efficiency and versatility are the main guiding notions for the development of new control approaches.

Bibliography

[1] A. Agudo, F. Moreno-Noguer, B. Calvo, and J. M. M. Montiel, "Sequential non-rigid structure from motion using physical priors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 5, pp. 979–994, 2016. DOI: 10.1109/TPAMI.2015.2469293 (cit. on p. 121).

[2] F. Alambeigi, Z. Wang, R. Hegeman, Y.-H. Liu, and M. Armand, "A robust data-driven approach for online learning and manipulation of unmodeled 3-D heterogeneous compliant objects," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4140–4147, 2018. DOI: 10.1109/LRA.2018.2863376 (cit. on pp. 28, 31, 37, 39, 42).

[3] F. Alambeigi, Z. Wang, R. Hegeman, Y.-H. Liu, and M. Armand, "Autonomous data-driven manipulation of unknown anisotropic deformable tissues using unmodelled continuum manipulators," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 254–261, 2019. DOI: 10.1109/LRA.2018.2888896 (cit. on p. 43).

[4] J. Alonso-Mora, S. Baker, and D. Rus, "Multi-robot formation control and object transport in dynamic environments via constrained optimization," *International Journal of Robotics Research*, vol. 36, no. 9, pp. 1000–1021, 2017. DOI: 10.1177/0278364917719333 (cit. on pp. 40, 73).

[5] J. Alonso-Mora, R. Knepper, R. Siegwart, and D. Rus, "Local motion planning for collaborative multi-robot manipulation of deformable objects," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 5495–5502. DOI: 10.1109/ICRA.2015.7139967 (cit. on pp. 30, 32, 35, 36, 38, 39, 42, 94).

[6] J. Alonso-Mora, E. Montijano, M. Schwager, and D. Rus, "Distributed multi-robot formation control among obstacles: A geometric and optimization approach with consensus," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 5356–5363. DOI: 10.1109/ICRA.2016.7487747 (cit. on p. 73).

[7] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *European Control Conference (ECC)*, 2019, pp. 3420–3431. DOI: 10.23919/ECC.2019.8796030 (cit. on pp. 81, 107).

[8] R. Antonova, J. Yang, P. Sundaresan, D. Fox, F. Ramos, and J. Bohg, “A bayesian treatment of real-to-sim for deformable object manipulation,” *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 5819–5826, 2022. DOI: 10.1109/LRA.2022.3157377 (cit. on p. 121).

[9] M. Aranda, J. A. Corrales Ramon, Y. Mezouar, A. Bartoli, and E. Ozgur, “Monocular visual shape tracking and servoing for isometrically deforming objects,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 7542–7549. DOI: 10.1109/IROS45743.2020.9341646 (cit. on p. 46).

[10] M. Aranda, G. Lopez-Nicolas, C. Sagues, and M. M. Zavlanos, “Distributed formation stabilization using relative position measurements in local coordinates,” *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 3925–3935, 2016. DOI: 10.1109/TAC.2016.2527719 (cit. on p. 39).

[11] M. Aranda, J. Sanchez, J. A. Corrales Ramon, and Y. Mezouar, “Robotic motion coordination based on a geometric deformation measure,” *IEEE Systems Journal*, vol. 16, no. 3, pp. 3689–3699, 2022. DOI: 10.1109/JSYST.2021.3107779 (cit. on pp. 72, 93, 95, 96, 98–101).

[12] V. E. Arriola-Rios and J. L. Wyatt, “A multimodal model of object deformation under robotic pushing,” *IEEE Transactions on Cognitive and Developmental Systems*, vol. 9, no. 2, pp. 153–169, 2017. DOI: 10.1109/TCDS.2017.2664058 (cit. on p. 120).

[13] P. Artaso and G. Lopez-Nicolas, “Volume estimation of merchandise using multiple range cameras,” *Measurement*, vol. 89, pp. 223–238, 2016. DOI: 10.1016/j.measurement.2016.04.005 (cit. on p. 44).

[14] Y. Asano, H. Wakamatsu, E. Morinaga, E. Arai, and S. Hirai, “Deformation path planning for manipulation of flexible circuit boards,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 5386–5391. DOI: 10.1109/IROS.2010.5651132 (cit. on pp. 37, 38, 42).

[15] C. Audet and J. E. Dennis, “Analysis of generalized pattern searches,” *SIAM Journal on Optimization*, vol. 13, no. 3, pp. 889–903, 2002. DOI: 10.1137/S1052623400378742 (cit. on p. 53).

[16] H. Bai and J. T. Wen, “Cooperative load transport: a formation-control perspective,” *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 742–750, 2010. DOI: 10.1109/TRO.2010.2052169 (cit. on pp. 29, 31, 33, 36, 38, 39, 42).

[17] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos, “Revisiting active perception,” *Autonomous Robots*, vol. 42, no. 2, pp. 177–196, 2018. DOI: 10.1007/s10514-017-9615-3 (cit. on p. 45).

[18] F. Basile, F. Caccavale, P. Chiacchio, J. Coppola, and C. Curatella, “Task-oriented motion planning for multi-arm robotic systems,” *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 5, pp. 569–582, 2012. DOI: 10.1016/j.rcim.2012.02.007 (cit. on p. 40).

[19] M. Bauza and A. Rodriguez, "A probabilistic data-driven model for planar pushing," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3008–3015. DOI: 10.1109/ICRA.2017.7989345 (cit. on p. 120).

[20] D. Berenson, "Manipulation of deformable objects without modeling and simulating deformation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 4525–4532. DOI: 10.1109/IROS.2013.6697007 (cit. on pp. 28, 37, 42).

[21] B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora, "Model predictive contouring control for collision avoidance in unstructured dynamic environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4459–4466, 2019. DOI: 10.1109/LRA.2019.2929976 (cit. on p. 73).

[22] R. Buchanan, T. Bandyopadhyay, M. Bjelonic, L. Wellhausen, M. Hutter, and N. Kottege, "Walking posture adaptation for legged robot navigation in confined spaces," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2148–2155, 2019. DOI: 10.1109/LRA.2019.2899664 (cit. on p. 72).

[23] O. Burchan Bayazit, J.-M. Lien, and N. M. Amato, "Probabilistic roadmap motion planning for deformable objects," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2002, pp. 2126–2133. DOI: 10.1109/ROBOT.2002.1014854 (cit. on p. 72).

[24] W. D. Callister and D. G. Rethwisch, *Materials science and engineering: an introduction*. John Wiley & sons New York, 2018 (cit. on p. 17).

[25] Z. Cao, N. Gu, J. Jiao, S. Nahavandi, C. Zhou, and M. Tan, "A novel geometric transportation approach for multiple mobile manipulators in unknown environments," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1447–1455, 2018. DOI: 10.1109/JSYST.2016.2581171 (cit. on p. 72).

[26] C.-Y. Chai, W.-H. Peng, and S.-L. Tsao, "Object rearrangement through planar pushing: A theoretical analysis and validation," *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 2703–2719, 2022. DOI: 10.1109/TRO.2022.3153785 (cit. on p. 120).

[27] M. Chen, Y. Tang, X. Zou, K. Huang, L. Li, and Y. He, "High-accuracy multi-camera reconstruction enhanced by adaptive point cloud correction algorithm," *Optics and Lasers in Engineering*, vol. 122, pp. 170–183, 2019. DOI: 10.1016/j.optlaseng.2019.06.011 (cit. on p. 43).

[28] A. Cherubini, J. Leitner, V. Ortenzi, and P. Corke, "Towards vision-based manipulation of plastic materials," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 485–490. DOI: 10.1109/IROS.2018.8594108 (cit. on p. 35).

[29] D. Cox and S. John, "A statistical method for global optimization," in *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 2, 1992, pp. 1241–1246. DOI: 10.1109/ICSMC.1992.271617 (cit. on p. 127).

[30] J. J. Craig, *Introduction to robotics: Mechanics and Control*, 3rd ed. Pearson Prentice Hall, 2005 (cit. on p. 77).

[31] Z. Cui, W. Ma, J. Lai, H. K. Chu, and Y. Guo, "Coupled multiple dynamic movement primitives generalization for deformable object manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5381–5388, 2022. DOI: 10.1109/LRA.2022.3156656 (cit. on p. 72).

[32] I. Cuiral-Zueco, G. Lopez-Nicolas, and H. Araujo, "Gripper positioning for object deformation tasks," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 963–969. DOI: 10.1109/ICRA46639.2022.9812304 (cit. on p. 94).

[33] D. Daney, Y. Papegay, and B. Madeline, "Choosing measurement poses for robot calibration with the local convergence method and tabu search," *The International Journal of Robotics Research*, vol. 24, no. 6, pp. 501–518, 2005. DOI: 10.1177/0278364905053185 (cit. on p. 78).

[34] P. Dang, F. L. Lewis, K. Subbarao, and H. Stephanou, "Shape control of flexible structure using potential field method," in *IEEE International Conference on Control Applications*, 2008, pp. 540–546. DOI: 10.1109/CCA.2008.4629650 (cit. on pp. 33, 38, 42).

[35] J. Das and N. Sarkar, "Autonomous shape control of a deformable object by multiple manipulators," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 62, no. 1, pp. 3–27, 2011. DOI: 10.1007/s10846-010-9436-5 (cit. on pp. 26, 32, 37, 38, 42, 43).

[36] F. M. De Rainville, J. P. Mercier, C. Gagne, P. Giguere, and D. Laurendeau, "Multisensor placement in 3D environments via visibility estimation and derivative-free optimization," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 3327–3334. DOI: 10.1109/ICRA.2015.7139658 (cit. on p. 45).

[37] A. Delgado, J. A. Corrales, Y. Mezouar, L. Lequievre, C. Jara, and F. Torres, "Tactile control based on Gaussian images and its application in bi-manual manipulation of deformable objects," *Robotics and Autonomous Systems*, vol. 94, pp. 148–161, 2017. DOI: 10.1016/j.robot.2017.04.017 (cit. on pp. 34, 42).

[38] J. Delmerico, S. Isler, R. Sabzevari, and D. Scaramuzza, "A comparison of volumetric information gain metrics for active 3D object reconstruction," *Autonomous Robots*, vol. 42, no. 2, pp. 197–208, 2018. DOI: 10.1007/s10514-017-9634-0 (cit. on p. 44).

[39] Z. Deng, A. Li, and X. Liu, "Equivalent virtual cameras to estimate a six-degree-of-freedom pose in restricted-space scenarios," *Measurement*, vol. 184, p. 109 903, 2021. DOI: 10.1016/j.measurement.2021.109903 (cit. on p. 44).

[40] Z. Deng, M. Stommel, and W. Xu, “A novel soft machine table for manipulation of delicate objects inspired by caterpillar locomotion,” *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 3, pp. 1702–1710, 2016. DOI: 10.1109/TMECH.2016.2519333 (cit. on p. 72).

[41] D. V. Dimarogonas and K. J. Kyriakopoulos, “A connection between formation infeasibility and velocity alignment in kinematic multi-agent systems,” *Automatica*, vol. 44, no. 10, pp. 2648–2654, 2008. DOI: 10.1016/j.automatica.2008.03.013 (cit. on p. 94).

[42] S. Duenser, J. M. Bern, R. Poranne, and S. Coros, “Interactive robotic manipulation of elastic objects,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 3476–3481. DOI: 10.1109/IROS.2018.8594291 (cit. on pp. 27, 42).

[43] E. Coumans, *Pybullet Real-Time Physics Simulation*, https://pybullet.org, 2010 (cit. on p. 128).

[44] H. Ebel, E. Sharafian Ardakani, and P. Eberhard, “Distributed model predictive formation control with discretization-free path planning for transporting a load,” *Robotics and Autonomous Systems*, vol. 96, pp. 211–223, 2017. DOI: 10.1016/j.robot.2017.07.007 (cit. on p. 94).

[45] H. Emadi, T. Gao, and S. Bhattacharya, “Visibility-based target-tracking game: Bounds and tracking strategies,” *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 1917–1924, 2017. DOI: 10.1109/LRA.2017.2714980 (cit. on p. 44).

[46] K. Fathian, T. H. Summers, and N. R. Gans, “Robust distributed formation control of agents with higher-order dynamics,” *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 495–500, 2018. DOI: 10.1109/LCSYS.2018.2841941 (cit. on pp. 94, 97).

[47] J. Felix-Rendon, J. C. Bello-Robles, and R. Q. Fuentes-Aguilar, “Control of differential-drive mobile robots for soft object deformation,” *ISA Transactions*, vol. 117, pp. 221–233, 2021. DOI: 10.1016/j.isatra.2021.01.057 (cit. on p. 94).

[48] Z. Feng, G. Hu, Y. Sun, and J. Soon, “An overview of collaborative robotic manipulation in multi-robot systems,” *Annual Reviews in Control*, vol. 49, pp. 113–127, 2020. DOI: 10.1016/j.arcontrol.2020.02.002 (cit. on p. 18).

[49] B. Frank, C. Stachniss, N. Abdo, and W. Burgard, “Using gaussian process regression for efficient motion planning in environments with deformable objects,” in *Proceedings of the 9th AAAI Conference on Automated Action Planning for Autonomous Mobile Robots*, 2011, pp. 2–7. DOI: 10.5555/2908675.2908676 (cit. on p. 121).

[50] Z. Gao, A. Elibol, and N. Y. Chong, “Zero moment two edge pushing of novel objects with center of mass estimation,” *IEEE Transactions on Automation Science and Engineering*, 2022. DOI: 10.1109/TASE.2022.3208739 (cit. on p. 120).

[51] J. M. Gere and B. J. Goodno, *Mechanics of materials*. Cengage learning, 2011 (cit. on p. 17).

[52] P. Guler, K. Pauwels, A. Pieropan, H. Kjellstrom, and D. Kragic, "Estimating the deformability of elastic materials using optical flow and position-based dynamics," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2015, pp. 965–971. DOI: 10.1109/HUMANOIDS.2015.7363486 (cit. on p. 44).

[53] L. Han, Y. Zhang, and H. Wang, "Vision-based contact point selection for the fully non-fixed contact manipulation of deformable objects," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4368–4375, 2022. DOI: 10.1109/LRA.2022.3149578 (cit. on p. 120).

[54] R. Hartley and A. Zisserman, "Multiple view geometry in computer vision," in Cambridge University Press, 2000 (cit. on p. 44).

[55] A. Hegde and D. Ghose, "Multi-UAV collaborative transportation of payloads with obstacle avoidance," *IEEE Control Systems Letters*, vol. 6, pp. 926–931, 2022. DOI: 10.1109/LCSYS.2021.3087339 (cit. on p. 73).

[56] D. Henrich and H. Worn, *Robot manipulation of deformable objects*. Springer Science & Business Media, 2012 (cit. on p. 17).

[57] R. Herguedas, M. Aranda, G. Lopez-Nicolas, C. Sagues, and Y. Mezouar, "Multi-robot control with double-integrator dynamics and control barrier functions for deformable object transport," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 1485–1491. DOI: 10.1109/ICRA46639.2022.9812378 (cit. on pp. 23, 95, 110).

[58] R. Herguedas, G. Lopez-Nicolas, R. Aragues, and C. Sagues, "Survey on multi-robot manipulation of deformable objects," in *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2019, pp. 977–984. DOI: 10.1109/ETFA.2019.8868987 (cit. on pp. 22, 43, 72).

[59] R. Herguedas, G. Lopez-Nicolas, and C. Sagues, "Multi-camera coverage of deformable contour shapes," in *IEEE International Conference on Automation Science and Engineering (CASE)*, 2019, pp. 1597–1602. DOI: 10.1109/COASE.2019.8843053 (cit. on pp. 22, 45).

[60] R. Herguedas, G. Lopez-Nicolas, and C. Sagues, "Sistema multi-cámara mínimo para percepción de formas deformables," in *VIII Jornada De Jóvenes Investigadores del I3A*, 2019. DOI: 10.26754/jji-i3a.003578 (cit. on p. 22).

[61] R. Herguedas, G. Lopez-Nicolas, and C. Sagues, "Experimental multi-camera setup for perception of dynamic objects," in *Workshop ROMADO, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 11 874–11 878 (cit. on pp. 22, 44).

[62] R. Herguedas, G. Lopez-Nicolas, and C. Sagues, "Collision-free transport of 2D deformable objects," in *International Conference on Control, Automation and Systems (ICCAS)*, 2021, pp. 430–435. DOI: 10.23919/ICCAS52745.2021.9650027 (cit. on pp. 23, 73, 77, 86, 87).

[63] R. Herguedas, G. Lopez-Nicolas, and C. Sagues, "Coverage of deformable contour shapes with minimal multi-camera system," *Measurement*, vol. 190, p. 110 693, 2022. DOI: 10.1016/j.measurement.2021.110693 (cit. on p. 22).

[64] R. Herguedas, G. Lopez-Nicolas, and C. Sagues, "Multirobot transport of deformable objects with collision avoidance," *IEEE Systems Journal*, vol. 17, no. 2, pp. 3224–3234, 2023. DOI: 10.1109/JSYST.2022.3213972 (cit. on pp. 23, 95).

[65] M. Higashimori, K. Yoshimoto, and M. Kaneko, "Active shaping of an unknown rheological object based on deformation decomposition into elasticity and plasticity," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 5120–5126. DOI: 10.1109/ROBOT.2010.5509462 (cit. on pp. 27, 35, 42).

[66] J. Hollerbach, W. Khalil, and M. Gautier, "Model Identification," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds., 1st Edition, Springer Berlin Heidelberg, 2008, ch. 14, pp. 321–344 (cit. on p. 78).

[67] J. Hu, W. Liu, H. Zhang, J. Yi, and Z. Xiong, "Multi-robot object transport motion planning with a deformable sheet," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9350–9357, 2022. DOI: 10.1109/LRA.2022.3191190 (cit. on p. 94).

[68] Y. R. Hu and G. Vukovich, "Active robust shape control of flexible structures," *Mechatronics*, vol. 15, no. 7, pp. 807–820, 2005. DOI: 10.1016/j.mechatronics.2005.02.004 (cit. on pp. 32, 42).

[69] Z. Hu, P. Sun, and J. Pan, "Three-dimensional deformable object manipulation using fast online Gaussian process regression," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 979–986, 2018. DOI: 10.1109/LRA.2018.2793339 (cit. on pp. 28, 32, 39, 42, 121).

[70] J. Huang, Y. Cai, X. Chu, R. H. Taylor, and K. W. S. Au, "Non-fixed contact manipulation control framework for deformable objects with active contact adjustment," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2878–2885, 2021. DOI: 10.1109/LRA.2021.3062302 (cit. on p. 120).

[71] K. Hunte and J. Yi, "Pose control of a spherical object held by deformable sheet with multiple robots," *IFAC-PapersOnLine*, vol. 55, no. 37, pp. 414–419, 2022, 2nd Modeling, Estimation and Control Conference MECC 2022. DOI: 10.1016/j.ifacol.2022.11.218 (cit. on p. 94).

[72] Interreg Sudoe Programme and European Regional Development Fund (ERDF), *Project CoMManDIA*, http://commandia.unizar.es/, 2018 (cit. on p. 18).

[73] M. Jagersand, O. Fuentes, and R. Nelson, "Experimental evaluation of uncalibrated visual servoing for precision manipulation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 1997, pp. 2874–2880. DOI: 10.1109/ROBOT.1997.606723 (cit. on p. 31).

[74] B. Jia, Z. Hu, J. Pan, and D. Manocha, “Manipulating highly deformable materials using a visual feedback dictionary,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 239–246. DOI: 10.1109/ICRA.2018.8461264 (cit. on pp. 27, 30, 37, 39, 42).

[75] J. Jiao, Z. Cao, N. Gu, S. Nahavandi, Y. Yang, and M. Tan, “Transportation by multiple mobile manipulators in unknown environments with obstacles,” *IEEE Systems Journal*, vol. 11, no. 4, pp. 2894–2904, 2017. DOI: 10.1109/JSYST.2015.2416215 (cit. on p. 72).

[76] P. Jimenez, “Survey on model-based manipulation planning of deformable objects,” *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 2, pp. 154–163, 2012. DOI: 10.1016/j.rcim.2011.08.002 (cit. on p. 25).

[77] W. Jing, J. Polden, P. Y. Tao, C. F. Goh, W. Lin, and K. Shimada, “Model-based coverage motion planning for industrial 3D shape inspection applications,” in *IEEE International Conference on Automation Science and Engineering (CASE)*, 2017, pp. 1293–1300. DOI: 10.1109/COASE.2017.8256278 (cit. on p. 44).

[78] A. Joubair, A. Tahan, and I. Bonev, “Performances of observability indices for industrial robot calibration,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 2477–2484. DOI: 10.1109/IROS.2016.7759386 (cit. on p. 78).

[79] F. F. Khalil and P. Payeur, “Dexterous robotic manipulation of deformable objects with multi-sensory feedback - a review,” in *Robot Manipulators Trends and Development*, A. Jimenez and B. M. A. Hadithi, Eds., InTech, 2010, pp. 587–621. DOI: 10.5772/9183 (cit. on p. 25).

[80] D. Kruse, R. J. Radke, and J. T. Wen, “Collaborative human-robot manipulation of highly deformable materials,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 3782–3787. DOI: 10.1109/ICRA.2015.7139725 (cit. on pp. 30, 35, 37, 39, 42).

[81] D. Kruse, R. J. Radke, and J. T. Wen, “Human-robot collaborative handling of highly deformable materials,” in *American Control Conference (ACC)*, 2017, pp. 1511–1516. DOI: 10.23919/ACC.2017.7963167 (cit. on p. 72).

[82] S. Kudoh, T. Gomi, R. Katano, T. Tomizawa, and T. Suehiro, “In-air knotting of rope by a dual-arm multi-finger robot,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 6202–6207. DOI: 10.1109/IROS.2015.7354262 (cit. on pp. 29, 42).

[83] J. D. Langsfeld, A. M. Kabir, K. N. Kaipa, and S. K. Gupta, “Robotic bimanual cleaning of deformable objects with online learning of part and tool models,” in *IEEE International Conference on Automation Science and Engineering (CASE)*, 2016, pp. 626–632. DOI: 10.1109/COASE.2016.7743460 (cit. on p. 119).

[84] J. D. Langsfeld, A. M. Kabir, K. N. Kaipa, and S. K. Gupta, "Integration of planning and deformation model estimation for robotic cleaning of elastically deformable objects," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 352–359, 2018. DOI: 10.1109/LRA.2017.2749280 (cit. on pp. 28, 30, 37, 39, 42).

[85] L. Latecki, R. Lakamper, and T. Eckhardt, "Shape descriptors for non-rigid shapes with a single closed contour," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2000, pp. 424–429. DOI: 10.1109/CVPR.2000.855850 (cit. on pp. 57–59).

[86] Y. Li, Y. Wang, Y. Yue, D. Xu, M. Case, S. F. Chang, E. Grinspun, and P. K. Allen, "Model-driven feedforward prediction for manipulation of deformable objects," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 4, pp. 1621–1638, 2018. DOI: 10.1109/TASE.2017.2766228 (cit. on pp. 26, 34, 37, 39, 42).

[87] Z. Lin, L. Wang, Z. Chen, M. Fu, and Z. Han, "Necessary and sufficient graphical conditions for affine formation control," *IEEE Transactions on Automatic Control*, vol. 61, no. 10, pp. 2877–2891, 2016. DOI: 10.1109/TAC.2015.2504265 (cit. on p. 94).

[88] P. Long, W. Khalil, and P. Martinet, "Robotic cutting of soft materials using force control & image moments," in *International Conference on Control Automation Robotics and Vision (ICARCV)*, 2014, pp. 474–479. DOI: 10.1109/ICARCV.2014.7064351 (cit. on pp. 29, 31, 35, 37, 39, 42).

[89] P. Long, W. Khalil, and P. Martinet, "Dynamic modeling of cooperative robots holding flexible objects," in *International Conference on Advanced Robotics (ICAR)*, 2015, pp. 182–187. DOI: 10.1109/ICAR.2015.7251453 (cit. on pp. 27, 42).

[90] G. Lopez-Nicolas, M. Aranda, and Y. Mezouar, "Adaptive multirobot formation planning to enclose and track a target with motion and visibility constraints," *IEEE Transactions on Robotics*, vol. 36, no. 1, pp. 142–156, 2020. DOI: 10.1109/TRO.2019.2943059 (cit. on p. 45).

[91] G. Lopez-Nicolas, R. Herguedas, M. Aranda, and Y. Mezouar, "Simultaneous shape control and transport with multiple robots," in *IEEE International Conference on Robotic Computing (IRC)*, 2020, pp. 218–225. DOI: 10.1109/IRC.2020.00042 (cit. on pp. 23, 72, 95).

[92] G. Lopez-Nicolas, E. Ozgur, and Y. Mezouar, "Parking objects by pushing using uncalibrated visual servoing," *Autonomous Robots*, vol. 43, pp. 1063–1078, 2019. DOI: 10.1007/s10514-018-9782-x (cit. on p. 120).

[93] P. C. Lusk, X. Cai, S. Wadhwania, A. Paris, K. Fathian, and J. P. How, "A distributed pipeline for scalable, deconflicted formation flying," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5213–5220, 2020. DOI: 10.1109/LRA.2020.3006823 (cit. on p. 94).

[94] D. Mcconachie and D. Berenson, “Estimating model utility for deformable object manipulation using multiarmed bandit methods,” *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 967–979, 2018. DOI: 10.1109/TASE.2018.2822669 (cit. on pp. 28, 36, 42).

[95] D. McConachie, A. Dobson, M. Ruan, and D. Berenson, “Manipulating deformable objects by interleaving prediction, planning, and control,” *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 957–982, 2020. DOI: 10.1177/0278364920918299 (cit. on p. 94).

[96] T. J. Mitchell, “An algorithm for the construction of “D-optimal” experimental designs,” *Technometrics*, vol. 16, no. 2, pp. 203–210, 1974. DOI: 10.2307/1267940 (cit. on p. 78).

[97] E. Moya-Lasheras and C. Sagues, “Run-to-run control with Bayesian optimization for soft landing of short-stroke reluctance actuators,” *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 6, pp. 2645–2656, 2020. DOI: 10.1109/TMECH.2020.2987942 (cit. on p. 121).

[98] S. Moza, “sahilm89/lhsmdu: Latin hypercube sampling with multi-dimensional uniformity (LHSMDU): Speed boost minor compatibility fixes,” 2020. DOI: 10.5281/zenodo.3929531 (cit. on p. 129).

[99] M. Mukadam, A. Borum, and T. Bretl, “Quasi-static manipulation of a planar elastic rod using multiple robotic grippers,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, pp. 55–60. DOI: 10.1109/IROS.2014.6942540 (cit. on p. 42).

[100] D. Navarro-Alarcon and Y. H. Liu, “Fourier-based shape servoing: A new feedback method to actively deform soft objects into desired 2-D image contours,” *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 272–279, 2018. DOI: 10.1109/TRO.2017.2765333 (cit. on pp. 27, 29, 34, 37, 42, 43).

[101] D. Navarro-Alarcon, Y. H. Liu, J. G. Romero, and P. Li, “Model-free visually servoed deformation control of elastic objects by robot manipulators,” *IEEE Transactions on Robotics*, vol. 29, no. 6, pp. 1457–1468, 2013. DOI: 10.1109/TRO.2013.2275651 (cit. on pp. 31, 32, 34, 37, 39, 42).

[102] D. Navarro-Alarcon, H. M. Yip, Z. Wang, Y. H. Liu, F. Zhong, T. Zhang, and P. Li, “Automatic 3-D manipulation of soft objects by robotic arms with an adaptive deformation model,” *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 429–441, 2016. DOI: 10.1109/TRO.2016.2533639 (cit. on pp. 27, 33, 34, 37, 39, 42).

[103] J. Nogueira, R. Martinez-Cantin, A. Bernardino, and L. Jamone, “Unscented Bayesian optimization for safe robot grasping,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 1967–1972. DOI: 10.1109/IROS.2016.7759310 (cit. on p. 121).

[104] E. Nuger and B. Benhabib, “Multi-camera active-vision for markerless shape recovery of unknown deforming objects,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 92, no. 2, pp. 223–264, 2018. DOI: 10.1007/s10846-018-0773-0 (cit. on pp. 45, 51).

[105] F. M. Nyikosa, M. A. Osborne, and S. J. Roberts, “Bayesian optimization for dynamic problems,” *arXiv:1803.03432*, 2018. DOI: 10.48550/arXiv.1803.03432 (cit. on pp. 123, 124).

[106] K.-K. Oh, M.-C. Park, and H.-S. Ahn, “A survey of multi-agent formation control,” *Automatica*, vol. 53, pp. 424–440, 2015. DOI: 10.1016/j.automatica.2014.10.022 (cit. on p. 93).

[107] C. Piciarelli, L. Esterle, A. Khan, B. Rinner, and G. L. Foresti, “Dynamic reconfiguration in camera networks: A short survey,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 5, pp. 965–977, 2016. DOI: 10.1109/TCSVT.2015.2426575 (cit. on p. 45).

[108] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. Cambridge, MA, USA: MIT Press, 2006 (cit. on p. 124).

[109] C. Robin and S. Lacroix, “Multi-robot target detection and tracking: Taxonomy and survey,” *Autonomous Robots*, vol. 40, no. 4, pp. 729–760, 2016. DOI: 10.1007/s10514-015-9491-7 (cit. on p. 44).

[110] F. Ruggiero, A. Petit, D. Serra, A. C. Satici, J. Cacace, A. Donaire, F. Ficuciello, L. R. Buonocore, G. A. Fontanelli, V. Lippiello, L. Villani, and B. Siciliano, “Nonprehensile manipulation of deformable objects: Achievements and perspectives from the robotic dynamic manipulation project,” *IEEE Robotics & Automation Magazine*, vol. 25, no. 3, pp. 83–92, 2018. DOI: 10.1109/MRA.2017.2781306 (cit. on pp. 32, 36, 39, 42, 43, 120).

[111] J. Sanchez, J. A. Corrales, B. C. Bouzgarrou, and Y. Mezouar, “Robotic manipulation and sensing of deformable objects in domestic and industrial applications: A survey,” *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 688–716, 2018. DOI: 10.1177/0278364918779698 (cit. on p. 25).

[112] D. S. Schacter, M. Donnici, E. Nuger, M. Mackay, and B. Benhabib, “A multi-camera active-vision system for deformable-object-motion capture,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 75, no. 3-4, pp. 413–441, 2014. DOI: 10.1007/s10846-013-9961-0 (cit. on p. 45).

[113] M. Schwager, B. J. Julian, M. Angermann, and D. Rus, “Eyes in the sky: Decentralized control for the deployment of robotic camera networks,” *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1541–1561, 2011. DOI: 10.1109/JPROC.2011.2158377 (cit. on p. 45).

[114] J. T. Shepard and C. A. Kitts, "A multirobot control architecture for collaborative missions comprised of tightly coupled, interconnected tasks," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1435–1446, 2018. DOI: 10.1109/JSYST.2016.2590430 (cit. on p. 72).

[115] M. Shetab-Bushehri, M. Aranda, Y. Mezouar, and E. Ozgur, "As-rigid-as-possible shape servoing," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3898–3905, 2022. DOI: 10.1109/LRA.2022.3145960 (cit. on p. 94).

[116] K. Simon and R. Basri, "Elasticity-based matching by minimising the symmetric difference of shapes," *IET Computer Vision*, vol. 12, no. 4, pp. 412–423, 2018. DOI: 10.1049/iet-cvi.2017.0277 (cit. on pp. 36, 38, 42).

[117] O. Sorkine and M. Alexa, "As-rigid-as-possible surface modeling," in *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, Eurographics Association, 2007, pp. 109–116. DOI: 10.2312/SGP/SGP07/109-116 (cit. on p. 111).

[118] E. Squires, P. Pierpaoli, and M. Egerstedt, "Constructive barrier certificates with applications to fixed-wing aircraft collision avoidance," in *IEEE Conference on Control Technology and Applications (CCTA)*, 2018, pp. 1656–1661. DOI: 10.1109/CCTA.2018.8511342 (cit. on p. 73).

[119] K. Sreenath and V. Kumar, "Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots," in *Robotics: Science and Systems IX*, 2013. DOI: 10.15607/RSS.2013.IX.011 (cit. on pp. 39, 42).

[120] R. Strauss, F. Isvoranu, and G. Elber, "Geometric multi-covering," *Computers and Graphics*, vol. 38, no. 1, pp. 222–229, 2014. DOI: 10.1016/j.cag.2013.10.018 (cit. on pp. 44, 51).

[121] J. Stuber, C. Zito, and R. Stolkin, "Let's push things forward: A survey on robot pushing," *Frontiers in Robotics and AI*, vol. 7, 2020. DOI: 10.3389/frobt.2020.00008 (cit. on p. 120).

[122] D. Sun, J. K. Mills, and Y. Liu, "Position control of robot manipulators manipulating a flexible payload," *The International Journal of Robotics Research*, vol. 18, no. 3, pp. 319–332, 1999. DOI: 10.1177/02783649922066231 (cit. on pp. 31, 42).

[123] A. M. Sundaram, W. Friedl, and M. A. Roa, "Environment-aware grasp strategy planning in clutter for a variable stiffness hand," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 9377–9384. DOI: 10.1109/IROS45743.2020.9340689 (cit. on p. 119).

[124] R. Tallamraju, D. H. Salunkhe, S. Rajappa, A. Ahmad, K. Karlapalem, and S. V. Shah, "Motion planning for multi-mobile-manipulator payload transport systems," in *IEEE International Conference on Automation Science and Engineering (CASE)*, 2019, pp. 1469–1474. DOI: 10.1109/COASE.2019.8842840 (cit. on pp. 72, 73).

[125] T. Tang, C. Liu, W. Chen, and M. Tomizuka, "Robotic manipulation of deformable objects by tangent space mapping and non-rigid registration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 2689–2696. DOI: 10.1109/IROS.2016.7759418 (cit. on pp. 29, 39, 42).

[126] T. Tang, C. Wang, and M. Tomizuka, "A framework for manipulating deformable linear objects by coherent point drift," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3426–3433, 2018. DOI: 10.1109/LRA.2018.2852770 (cit. on pp. 29, 34, 39, 42, 43).

[127] Y. Tang, L. Li, C. Wang, M. Chen, W. Feng, X. Zou, and K. Huang, "Real-time detection of surface deformation and strain in recycled aggregate concrete-filled steel tubular columns via four-ocular vision," *Robotics and Computer-Integrated Manufacturing*, vol. 59, pp. 36–46, 2019. DOI: 10.1016/j.rcim.2019.03.001 (cit. on p. 43).

[128] H. G. Tanner, S. G. Loizou, and K. J. Kyriakopoulos, "Nonholonomic navigation and control of cooperating mobile manipulators," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 53–64, 2003. DOI: 10.1109/TRA.2002.807549 (cit. on p. 72).

[129] S. Tokumoto, Y. Fujita, and S. Hirai, "Deformation modeling of viscoelastic objects for their shape control," in *IEEE International Conference on Robotics and Automation (ICRA)*, 1999, pp. 767–772. DOI: 10.1109/ROBOT.1999.770067 (cit. on p. 42).

[130] T. Wada, S. Hirai, S. Kawamura, and N. Kamiji, "Robust manipulation of deformable objects by a simple PID feedback," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2001, pp. 85–90. DOI: 10.1109/ROBOT.2001.932534 (cit. on pp. 28, 32, 38, 42).

[131] F. Wang, Z. Qian, Z. Yan, C. Yuan, and W. Zhang, "A novel resilient robot: Kinematic analysis and experimentation," *IEEE Access*, vol. 8, pp. 2885–2892, 2020. DOI: 10.1109/ACCESS.2019.2962058 (cit. on p. 72).

[132] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, 2017. DOI: 10.1109/TRO.2017.2659727 (cit. on pp. 73, 80, 81, 95, 107).

[133] Z. Wang and M. Schwager, "Kinematic multi-robot manipulation with no communication using force feedback," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 427–432, 2016. DOI: 10.1109/ICRA.2016.7487163 (cit. on p. 39).

[134] S. Wilson, P. Glotfelter, L. Wang, S. Mayya, G. Notomista, M. Mote, and M. Egerstedt, "The Robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems," *IEEE Control Systems Magazine*, vol. 40, no. 1, pp. 26–44, 2020. DOI: 10.1109/MCS.2019.2949973 (cit. on pp. 90, 114).

[135] C. Yang, G. N. Sue, Z. Li, L. Yang, H. Shen, Y. Chi, A. Rai, J. Zeng, and K. Sreenath, "Collaborative navigation and manipulation of a cable-towed load by multiple quadrupedal robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 041–10 048, 2022. DOI: 10.1109/LRA.2022.3191170 (cit. on p. 94).

[136] H. Yin, A. Varava, and D. Kragic, "Modeling, learning, perception, and control methods for deformable object manipulation," *Science Robotics*, vol. 6, no. 54, eabd8803, 2021. DOI: 10.1126/scirobotics.abd8803 (cit. on p. 72).

[137] T. Zhang, J. Liu, S. Liu, C. Tang, and P. Jin, "A 3D reconstruction method for pipeline inspection based on multi-vision," *Measurement*, vol. 98, pp. 35–48, 2017. DOI: 10.1016/j.measurement.2016.11.004 (cit. on pp. 44, 46).

[138] X. Zhang, X. Chen, J. L. Alarcon-Herrera, and Y. Fang, "3-D model-based multi-camera deployment: A recursive convex optimization approach," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 6, pp. 3157–3169, 2015. DOI: 10.1109/TMECH.2015.2411593 (cit. on p. 44).

[139] S. Zhao, "Affine formation maneuver control of multiagent systems," *IEEE Transactions on Automatic Control*, vol. 63, no. 12, pp. 4140–4155, 2018. DOI: 10.1109/TAC.2018.2798805 (cit. on p. 94).

[140] S. Zhao, D. V. Dimarogonas, Z. Sun, and D. Bauso, "A general approach to coordination control of mobile agents with motion constraints," *IEEE Transactions on Automatic Control*, vol. 63, no. 5, pp. 1509–1516, 2018. DOI: 10.1109/TAC.2017.2750924 (cit. on p. 94).

[141] J. Zhu, A. Cherubini, C. Dune, D. Navarro-Alarcon, F. Alambeigi, D. Berenson, F. Ficuciello, K. Harada, J. Kober, X. Li, J. Pan, W. Yuan, and M. Gienger, "Challenges and outlook in robotic manipulation of deformable objects," *IEEE Robotics & Automation Magazine*, vol. 29, no. 3, pp. 67–77, 2022. DOI: 10.1109/MRA.2022.3147415 (cit. on p. 71).

[142] J. Zhu, B. Navarro, P. Fraisse, A. Crosnier, and A. Cherubini, "Dual-arm robotic manipulation of flexible cables," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 479–484. DOI: 10.1109/IROS.2018.8593780 (cit. on pp. 28, 29, 37, 39, 42).