UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

DIPARTIMENTO DI SCIENZE E METODI DELL'INGEGNERIA

Corso di Laurea Magistrale in Ingegneria Meccatronica

THE KILOBOT SOFT ROBOT: A SOFT-BODIED MODULAR RECONFIGURABLE ROBOTIC SYSTEM

IL SOFT ROBOT KILOBOT: UN SISTEMA ROBOTICO DEFORMABILE, MODULARE E RICONFIGURABILE

Relatore:

Prof. Lorenzo Sabattini

Tesi di Laurea di:

Federico Pratissoli

Correlatori esterni:

Prof. Roderich Groß

Dott. Ing. Andreagiovanni Reina

Dott. Ing. Yuri Kaszubowski Lopes

Anno Accademico 2017 - 2018

To my family

Il Soft Robot Kilobot: Un Sistema Robotico Deformabile, Modulare e Riconfigurabile

Riassunto in lingua italiana

Federico Pratissoli

Sommario

Lo studio qui presentato deriva da una collaborazione di due importanti aree di ricerca in robotica: la swarm robotics e la soft robotics. La swarm intelligence è diventata oggigiorno importante per i numerosi studi in ambito di controllo di un grande numero di robot attraverso algoritmi di controllo decentralizzati e distribuiti. L'idea si basa sul concetto che una cooperazione di molti robot permette di portare a termine compiti e obiettivi che il singolo semplice dispositivo non è in grado di svolgere. In parallelo la soft robotics accresce la sua importanza: area di ricerca nata grazie alla nascita e al progresso di nuovi materiali deformabili o con rigidità controllabile. Il progetto qui descritto nasce come una relazione individuata tra questi due ambiti di studi.

La tesi presenta il Soft Robot Kilobot, una nuova tipologia di robots con struttura deformabile caratterizzato dalle proprietà di modularità e riconfigurabilità. Il Robot è realizzato connettendo tra loro, per mezzo di un materiale elastico, un definito gruppo di robots mobili di piccole dimensioni. I dispositivi usati sono i Kilobot, progettati da K-team e disponibili sul mercato. Il robot realizzato è deformabile, composto da più moduli indipendenti e dotato di completa mobilità autonoma.

Ogni modulo si muove per mezzo di due piccoli motori a vibrazioni, è in grado di stimare la distanza di un robot vicino e trasmettere o ricevere byte di messaggi. Algoritmi distribuiti sono stati progettati al fine di controllare lo spostamento e la deformazione del Soft Robot Kilobot. Questi controlli si basano sulla comunicazione e sulla trasmissione di informazioni tra un modulo e l'altro. A questi viene implementata la capacità di localizzarsi relativamente ai robot vicini in una zona del Soft

Robot. Lo spostamento del sistema complessivo è quindi ottenuto tramite un controllo sui singoli elementi che lo costituiscono, i quali prendono decisioni durante il moto basandosi sulle distanze misurate e le informazioni ricevute tra i robot vicini.

Diversi esperimenti sono stati eseguiti al fine di validare le caratteristiche e le capacità del Robot realizzato, tra queste: l'abilità del sistema di seguire una traiettoria predefinita, la capacità di questo di deformare la propria geometria, da una forma estesa regolare e reticolare ad una compressa e allungata, e infine la proprietà del controllo in grado di ottenere uno spostamento regolare del Soft Robot indipendentemente dal numero di moduli che lo costituiscono. L'ultima caratteristica è stata testata con una serie di esperimenti nei quali la dimensione del sistema veniva progressivamente aumentata, fino al valore massimo di 49 robot connessi tra loro.

Al meglio delle nostre conoscenze, considerando il numero di moduli usati per gli esperimenti, questa è la prima rigorosa valutazione sperimentale di un sistema completamente attuato, modulare e riconfigurabile di tali dimensioni, e il più grande robot a struttura deformabile modulare dotato di mobilità propria.

In questa tesi sono analizzati i risultati sperimentali tra i quali il coefficiente di distorsione della forma, la deviazione durante il movimento dalla traiettoria desiderata e in generale la abilità di seguire un percorso predefinito. Alcune conclusioni sono state trattate in seguito a tali risultati: la precisione e accuratezza nel movimento del Soft Robot Kilobot aumenta con l'aumentare della dimensione del robot e quindi col numero di moduli che lo costituiscono. Inoltre è stato visto come il sistema sia in grado di ripercorrere il percorso desiderato mantenendo approssimativamente una velocità costante. Sorge spontanea la domanda e successivamente l'ipotesi, sostenuta in questo progetto, che per un robot modulare, che effettivamente si muove sul suolo, una struttura deformabile, con connessioni elastiche, sia superiore ad una rigida, migliorando il sistema grazie caratteristiche aggiuntive come la deformazione controllata della forma.

Capitolo 1

Introduzione

L'idea di questo progetto nasce dalla compartecipazione di due importanti ambiti della robotica: la Swarm Robotics e la Soft Robotics. Il primo si tratta di uno studio ispirato dal comportamento collettivo di insetti, come formiche o api, con l'obiettivo di utilizzare algoritmi di controllo distribuito per spingere lo sciame a collaborare per risolvere un determinato obiettivo. Il secondo è incentrato sulla realizzazione di robots tramite materiali deformabili o flessibili. Anche in questo caso l'ispirazione viene dalla natura e in particolare da animali come vermi o polipi. Senza una struttura rigida questi sistemi sono permettono una maggiore flessibilità, adattabilità e un alto grado di robustezza e versatilità nello portare a termine compiti di varia natura. Inoltre, sono preferiti in applicazioni nelle quali l'interazione con l'essere umano è importante, essendo dotati di superfici morbide sono più sicuri.

L'idea è quella di realizzare un prototipo di robot a corpo deformabile non controllato da un singolo processore ma costituito di più moduli o elementi interconnessi e comunicanti tra loro che collaborando sono in grado di attuare il Soft Robot. Algoritmi di controllo distribuiti ispirati alla intelligenza collettiva e caratterizzati dalle proprietà di robustezza e scalabilità tipiche della Swarm robotics.

Il prototipo realizzato e studiato durante questo progetto è il Soft Robot Kilobot, un sistema robotico deformabile, modulare e riconfigurabile. I moduli sono indipendenti e completamente attuati e sono realizzati tramite il dispositivo Kilobot, questo si tratta di un semplice, poco costoso e facilmente ottenibile robot tipicamente usato in Swarm Robotics.

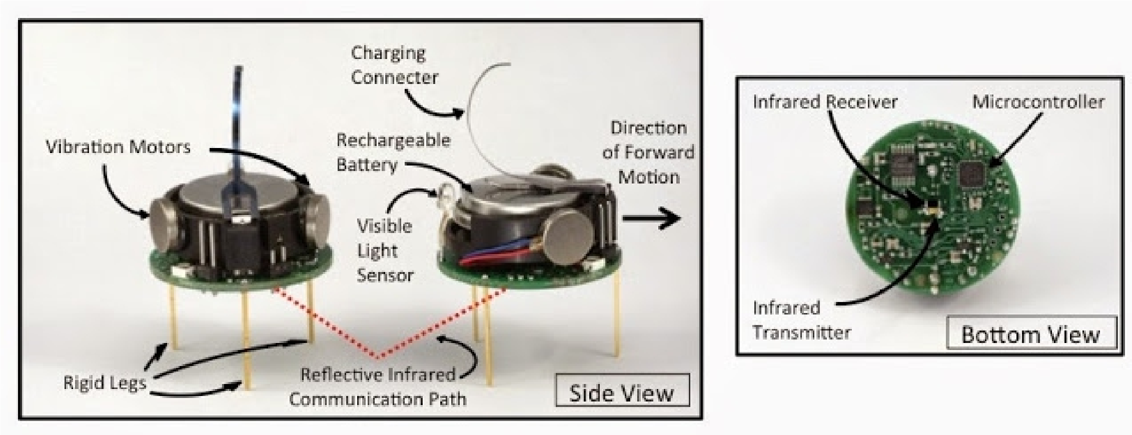


Figura 1.1: Diverse viste del Kilobot e dei suoi componenti hardware.

I Kilobot sono dispositivi dall'hardware semplice col fine di abbattere i costi permettendo sperimentazioni con centinaia di robot. Come mostrato in Figura 1.1, sono costituiti da un corpo circolare di raggio 3.3 cm e da tre gambe rigide che svolgono la sola funzione di supporto e alzano il robot fino a 3.4 cm. Tra gli altri componenti abbiamo: un connettore per caricare la batteria, due motori a vibrazione, un sensore per stimare il livello di luminosità nell'ambiente, un LED RGB, una batteria al litio ricaricabile, e un trasmettitore e ricevitore di segnali a infrarossi.

Il movimento di questi robot è basato sull'effetto slip-stick ottenuto grazie alla vibrazione controllata dei due motori. A seconda della superficie utilizzata e del livello della batteria il Kilobot ha una velocità massima di 1 cm/s in traslazione e 45 deg/s in rotazione. Trasmettitore e ricevitore IR sono situati al di sotto del corpo circolare, la comunicazione del messaggio avviene per rimbalzo del segnale sulla superficie, come mostrato in figura. In questo modo la trasmissione non è indirizzata e un messaggio è ricevuto da tutti i vicini nell'intorno del robot trasmettente. Inoltre l'intensità del segnale a infrarossi ricevuto è usata per stimare la distanza dal robot da cui proviene il segnale. Un messaggio può contenere al massimo 9 byte di dati ed è trasmesso con una frequenza di due messaggi ogni secondo.

Data la sua semplicità il Kilobot presenta alcune problematiche in termini di prestazioni: lo spostamento, l'aspetto più critico, è inaccurato, impreciso, e influenzato da una importante presenza di rumore, essendo basato sulla vibrazione. Il dispositivo inoltre non presenta nessun sensore che gli dia un feedback sullo stato attuale. Quindi diventa fondamentale la comunicazione e lo scambio delle informazioni con i robot vicini, tramite la quale è possibile generare una sorta di feedback e ottenere un controllo sulla movimentazione del dispositivo. Tuttavia, la trasmissione del segnale non è completamente affidabile, è influenzata, infatti, da numerose variabili: il tipo di superficie (sulla quale il segnale rimbalza), il livello di carica della batteria, collisioni tra segnali si possono verificare, soprattutto quando numerosi dispositivi sono in posizioni ravvicinate. Un buon software è stato necessario progettare al fine di compensare queste problematiche introdotte dal Kilobot.

Per la realizzazione del Soft Robot, più moduli sono collegati tra loro attraverso materiale elastico, in particolare molle elicoidali, formando una geometria regolare e reticolare. In questo modo la dimensione del sistema può essere arbitrariamente cambiata, a seconda, per esempio, del compito che deve essere svolto o dell'ambiente e condizioni in cui deve operare.

Algoritmi di controllo distribuito sono progettati per aggiungere al Soft Robot Kilobot la capacita di seguire la traiettoria desiderata, di contrarsi o estendersi deformando la propria forma quando richiesto e di mantenere la propria forma regolare e a reticolo durante uno spostamento indipendentemente dal numero di moduli che costituiscono il sistema. Al fine di validare e testare queste proprietà, numerosi esperimenti sono stati eseguiti su Soft Robot di diverse dimensioni, fino a quella massima di un Robot composto da 49 Kilobot.

La presente dissertazione è organizzata come segue: nel capitolo seguente vengono illustrati l'hardware progettato per realizzare il Soft Robot e gli algoritmi decentralizzati implementati su ogni Kilobot al fine di controllare il sistema. Il capitolo 3 descrive gli esperimenti eseguiti per mostrare e validare le abilità del robot costruito. E, infine, un ultimo capitolo tratta delle conclusioni raggiunte in seguito a questo studio e dei progetti futuri che possono essere di interesse per continuare l'attività di ricerca.

Capitolo 2

Il Soft Robot Kilobot

Come precedentemente menzionato i moduli che costituiscono il Soft Robot sono connessi tra loro da delle molle di forma elicoidale in materiale plastico. Ogni Kilobot in questo modo non è completamente libero di muoversi o ruotare ma è vincolato da queste connessioni fisiche. Queste limitazioni aiutano a compensare la notevole inaccuratezza nel movimento del dispositivo e aiutano a semplificare l'algoritmo di controllo.

Come mostrato in figura, il sistema è realizzato connettendo tra loro i Kilobot in una struttura reticolare. I Kilobot non sono adibiti per essere connessi a degli oggetti esterni, è stato, quindi, progettato uno scheletro con forma ad anello e costruito tramite stampa tridimensionale. Questo si incastra sulla batteria del robot, evitando di influenzare in alcun modo la movimentazione del modulo, i suoi motori e soprattutto evitando di coprire il LED RGB, componente importate per il sistema di rilevazione e tracciamento dei Kilobot (descritto nei capitoli seguenti).

I moduli, migliorati da questo scheletro esterno, sono completamente intercambiabili, e il sistema può essere arbitrariamente riconfigurato. La dimensione del Soft Robot può essere aumentata o diminuita senza condizionare le sue capacità e senza dover apportare cambiamenti al "codice". Questo è reso possibile dalla progettazione di algoritmi di controllo distribuiti che arricchiscono il robot realizzato delle proprietà di scalabilità, robustezza e modularità.

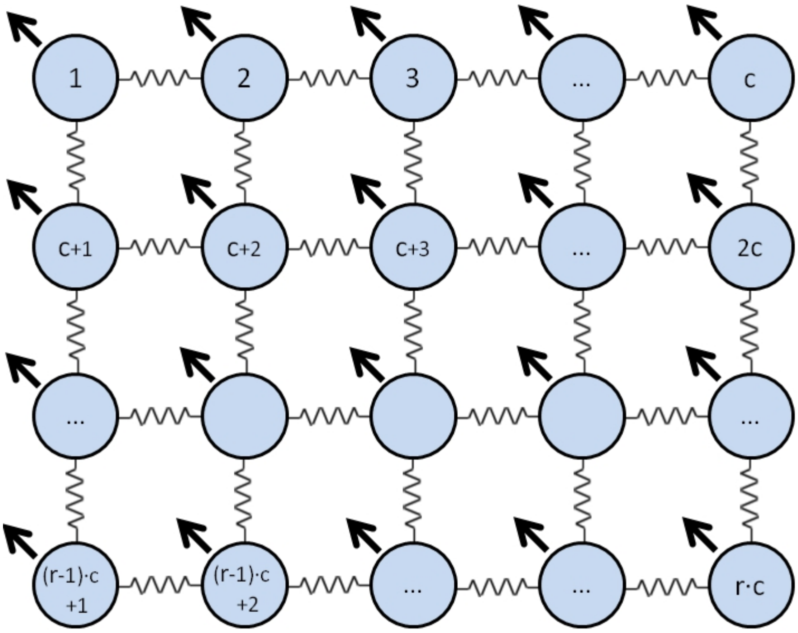


Figura 2.1: Il Soft Robot è costituito da moduli Kilobot connessi tra loro da molle, formando una struttura reticolare $r \times c$. Ogni modulo, identificato univocamente da un numero, è disposto ordinatamente lungo le file del reticolo.

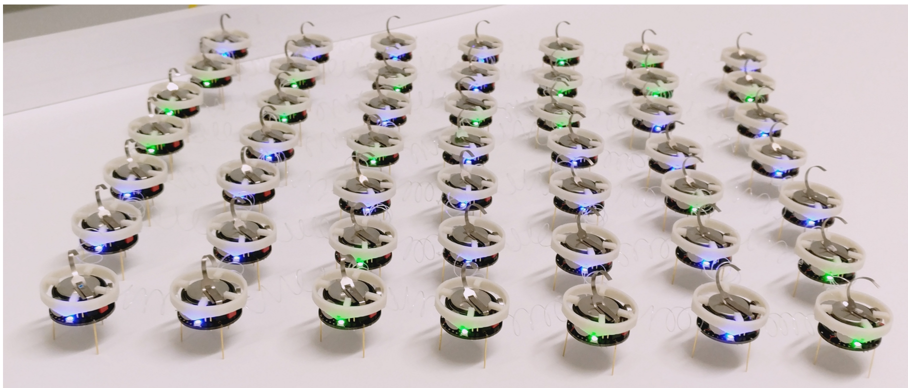


Figura 2.2: Un Soft Robot Kilobot 7×7 composto da 49 moduli connessi da 84 molle leggere. In questa foto i LED stanno mostrando il livello di carica della batteria di ciascun modulo. La batteria è considerata scarica se i colori mostrati sono il giallo o il rosso, altrimenti è considerata carica.

L'algoritmo è suddiviso in due parti principali: un sotto algoritmo nel quale viene trovata una soluzione al problema di localizzazione del singolo modulo tra altri e un secondo sotto algoritmo nel quale è implementato il controllo del movimento di ogni singolo modulo e quindi anche dell'intero sistema.

La localizzazione consiste nel determinare la posizione, relativa al modulo che sta eseguendo l'algoritmo, di ognuno dei robot vicini vincolati attorno ad esso dalla struttura reticolare. Guardando la Figura 2.1, il numero di "vicini" varia da un massimo di 8, quando il modulo è al centro del reticolo, ad un minimo di 3, quando è posizionato su un angolo del reticolo. L'algoritmo di localizzazione è basato su una efficiente e rapida trasmissione, condivisione e salvataggio delle distanze misurate dai robot vicini e sul calcolo dei rispettivi angoli. Grazie, infatti, ai vincoli imposti dalla struttura reticolare e all'ipotesi che a tutti i Kilobot viene assegnato un numero identificativo in maniera ordinata, è possibile localizzare univocamente un terzo modulo rispetto ad altri due moduli posizionati vicini ad esso. Tre robot sono, infatti, sufficienti per il calcolo di tutti gli angoli interni e, quindi, per localizzarsi a vicenda. E, le ambiguità che sorgerebbero in una situazione simile vengono eliminate dal modo in cui il Soft Robot è realizzato.

L'algoritmo di controllo del movimento utilizza i valori di distanze e angoli interni del reticolo, i quali sono aggiornati continuamente, in parallelo al movimento del Kilobot, nell'algoritmo di localizzazione. L'obiettivo è quello di muovere il Soft Robot Kilobot verso la direzione desiderata mantenendo la sua forma originale e reticolare costante. Il controllo progettato permette quindi di ottenere uno spostamento regolare e coerente del gruppo di Kilobot e di gestire la forma del sistema durante il movimento. Il movimento del singolo Kilobot è controllato tramite tre parametri principali: gli angoli, che vengono tenuti entro un intervallo centrato a 90 gradi (gli angoli interni del reticolo sono tutti a 90 gradi), le distanze laterali che impediscono collisioni con di robot vicini e le distanze diagonali che facilitano la gestione della forma (in questo caso estesa).

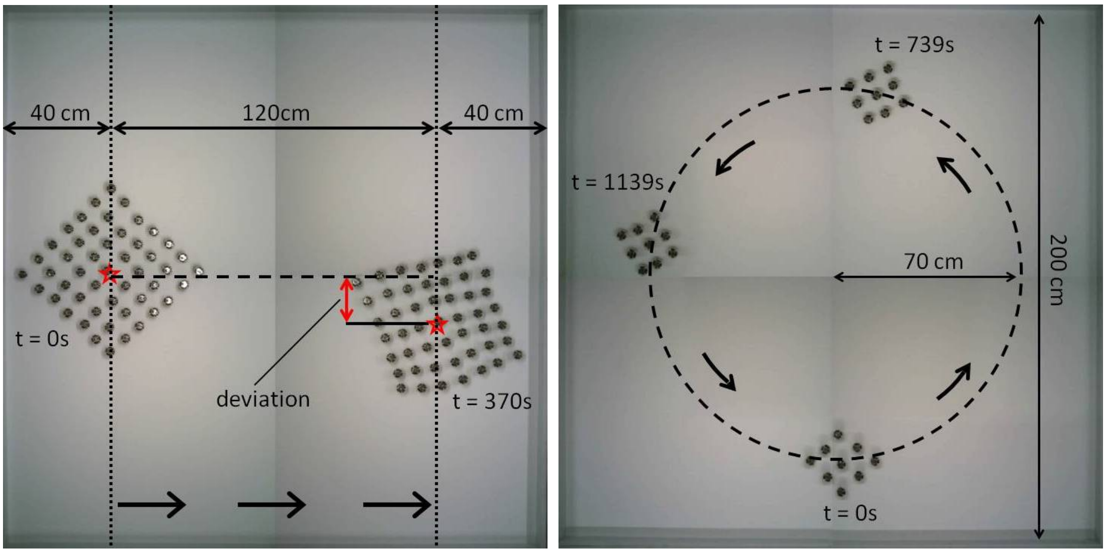
Per ottenere invece una deformazione della struttura da una forma reticolare estesa ad una romboidale compressa, l'algoritmo di controllo è lo stesso ma cambiano i parametri utilizzati: gli angoli sono tenuti a valori più alti (intorno ai 130 gradi) o più bassi (intorno ai 40 gradi) a seconda della posizione del Kilobot nel reticolo, il limite minimo per le distanze diagonali e laterali è diminuito in modo da permettere una compressione della forma.

Capitolo 3

Esperimenti

Al fine di valutare le capacità del Soft Robot Kilobot e di testare gli algoritmi implementati, diversi esperimenti sono stati condotti. In particolare tre tipologie di questi sono state ideate: un primo set di sperimentazioni mirato a testare la scalabilità del sistema, un secondo mirato a valutare la abilità del Robot di seguire un percorso predefinito e infine, uno specifico per testare la deformabilità della struttura.

Il primo esperimento consiste in una analisi delle prestazioni del Robot sottoposto ad una variazione della sua dimensione S, dove $S \in \{1\times 1, 2\times 2, \ldots, 7\times 7\}$, e, quindi, del numero di moduli che lo costituiscono. Il sistema è controllato per muoversi lungo una traiettoria rettilinea in assenza di un feedback esterno, posizione di partenza e di arrivo sono state definite prima di iniziare l'esperimento considerando il centro di massa del Robot come punto di riferimento. La prova viene considerata un successo se il Soft Robot è in grado di attraversare il traguardo nonostante l'errore crescente sulla traiettoria seguita (controllo ad anello aperto), viene, invece, considerata un fallimento altrimenti. Dai risultati di questo esperimento si è notato che aumentando il numero di moduli e, quindi, la dimensione S, l'accuratezza del sistema aumenta, così come la probabilità che questo raggiunga correttamente la linea di traguardo. In altre parole, la deviazione laterale dalla traiettoria desiderata che si genera durante lo spostamento diminuisce all'aumentare di S. In sistemi di maggiori dimensioni l'errore sulla movimentazione introdotto da ogni singolo modulo viene compensato dalla collaborazione collettiva.



(a) (b)

Figura 3.1: Le impostazioni dei due primi esperimenti: (a) posizione iniziale e finale di un 7×7 Soft Robot, sono evidenziati centro di massa e deviazione dalla traiettoria ideale; (b) posizioni, in tre istanti di tempo diversi, di un 3×3 Soft Robot mentre segue il percorso circolare

Il secondo esperimento valuta le prestazioni del Soft Robot Kilobot nel seguire una traiettoria curva predefinita in presenza in un feedback esterno (controllo ad anello chiuso) e nel mantenere in contemporanea la forma reticolare costante. Al sistema viene comandato di seguire un percorso circolare tenendo una velocità costante. Quattro prove su cinque sono state un successo in quanto il Robot è stato in grado di completare una rivoluzione con notevole accuratezza rispetto alla traiettoria desiderata.

Il terzo e ultimo esperimento valuta le prestazioni del Soft Robot Kilobot nel deformare su comando la sua geometria mentre segue una traiettoria rettilinea. Come in precedenza è implementato un controllo ad anello chiuso per dirigere il Robot sul percorso desiderato. In questo esperimento esistono 3 fasi: una iniziale, durante la quale il sistema di muove con direzione rettilinea mantenendo la forma estesa reticolare, una intermedia, durante la quale al sistema viene comandato di comprimersi e tale forma viene mantenuta fino alla fase finale, durante la quale il sistema, muovendosi in maniera rettilinea, riporta la sua forma nelle condizioni originali (a reticolo). I risultati mostrano che 10 prove su 11 hanno avuto successo.

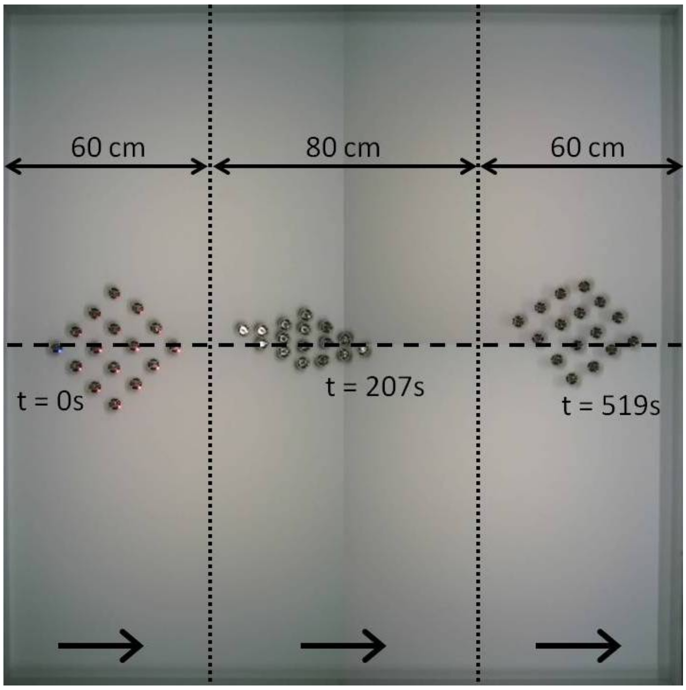


Figura 3.2: L'impostazione dell'ultimo esperimento: le due linee tratteggiate verticali delimitano le zone di spazio nelle quali, rispettivamente da sinistra, il Robot ha forma estesa, compressa e nuovamente estesa durante lo spostamento. La linea tratteggiata centrale indica la traiettoria seguita dal sistema in controllo ad anello chiuso

Capitolo 4

Conclusioni

In questa tesi presentiamo il Soft Robot Kilobot, una nuova idea per un robot modulare a corpo deformabile. Questo è composto da semplici unità mobili, i Kilobot, connesse da molle al fine di creare una struttura reticolare 2D deformabile. Inoltre, i moduli che costituiscono il sistema possono essere manualmente riconfigurabili per creare reticoli di diverse dimensioni. Gli algoritmi di controllo distribuiti progettati permettono al Soft Robot di muoversi su un piano, anche seguendo una traiettoria desiderata, e di controllare la propria forma, da reticolare ed estesa a compressa e romboidale. I risultati ottenuti dagli esperimenti mostrano le abilità del sistema anche in rapporto al numero di moduli che lo costituiscono. Gli errori durante lo spostamento introdotti da ciascun Kilobot sono infatti compensati in sistemi di grandi dimensioni, i quali, di conseguenza, si muovo con maggiore accuratezza.

Concludendo, questo studio presenta una delle prime implementazioni di robot a corpo deformabile composto da un grande numero di unità mobili completamente autonome e che può essere arbitrariamente riconfigurato. Inoltre, in questo progetto, è stata eseguita una delle prime rigorose valutazioni sperimentali di un robot modulare mobile di grandi dimensioni (fino a 49 unità).

Quindi, alla luce di tali risultati, nasce la domanda se le connessioni deformabili sono superiori a quelle rigide per robot modulari che si muovo sul suolo. Questo è un possibile problema che può essere investigato in attività future.

I risultati ottenuti e analizzati durante l'attività, illustrata nella presente tesi, hanno permesso la scrittura di un articolo scientifico in collaborazione con alcuni ricercatori dell'Università di Sheffield. L'articolo è stato mandato in occasione dello *special issue* "Soft Robotic Modeling and Control: Bringing Together Articulated Soft Robots and Soft-Bodied Robots" al giornale IJRR (International Journal of Robotics Research), ed ha il titolo: Federico Pratissoli, Andreagiovanni Reina, Yuri Kaszubowski Lopes, Lorenzo Sabattini, Roderich Groß (2019) The Kilobot Soft Robot: A Soft-Bodied Modular Reconfigurable Robotic System.

The Kilobot Soft Robot: A Soft-Bodied Modular Reconfigurable Robotic System

Federico Pratissoli

Abstract

Swarm intelligence has become nowadays an important area of research thanks to the ability of control of a large number of robots that have as basic requirement the communication between them. Therefore, it is possible to perform complex tasks, which the single simple robot could not perform, thanks to the cooperation between many of these simple robots. In the same way, soft robotics is a specific field of robotics that is gaining importance in the scientific research, born thanks the introduction of new materials and deformable connections. The project here described is a research that came out of the relationships seen between the two fields of robotics mentioned above. In this dissertation the Kilobot Soft Robot is proposed, a novel soft-bodied robot that is modular and reconfigurable. The Kilobot Soft Robot is realized by inter-connecting a group of miniature mobile robots, the commercially available Kilobot, through an elastic material. Hence, a deformable system composed by single modules is obtained, able to emulate the behaviour of a soft robot. The robot is fully autonomous. Each module can move using vibration motors, detect the distance of a neighbour and communicate messages among the neighbourhood. Distributed algorithms are designed to control the motion and the shape of the Kilobot Soft Robot. This control is based on the communication between the inter-connected elements, capable to self-localise, move and make decisions using only local information. Different experiments are conducted to prove the algorithms and to show the ability of the system to follow a defined trajectory, to squeeze and extend its shape and to have a control on its motion independently from the number of modules that constitute the Robot. The last property is validated through a series

of tests with up to 49 physical modules. To the best of our knowledge, considering the number of modules used, this is the first rigorous experimental evaluation of a self-propelling modular reconfigurable robot of this size, and the biggest soft-bodied modular robot to date made of fully autonomous modules. This dissertation shows the analysis of the results regarding the shape distortion, the deviation from the ideal path and the ability to follow a trajectory. We can say from these results that the motion accuracy of the Kilobot Soft Robot increases with the number of modules and that the Kilobot Soft Robot can move roughly with a constant speed following a path. We conclude with the hypothesis that, for a modular robot to effectively move across the ground, soft connections might be superior to rigid ones, enhancing the system with some advantages as shape deformation.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background and Motivation

A robot is a general machine capable of performing a certain task with different levels of accuracy, reliability or independence with the aim to help or replace human work. In the field of robotics, the 'rigid' or 'stiff' robots are very common. They are made of rigid structures like metal frames and their motion is determined by a relative movement between these rigid connections. They consist of a wide range of systems used in industry or in research for manufacturing, building, handling of heavy and dangerous materials, or working in extreme environmental conditions. These type of robots generally are very popular because they are able to perform repetitive motions over a long period of time with high reliability and speed. Automotive industry or automated warehouses are examples where these robot are frequently used. However, there are some situations where these robots can not be used, such as tasks that can not be performed because of the constraints imposed by a rigid structure or operations in which the collaboration with the human is important. Indeed a person, who works together with an automatic machine, which is moving in the space, constituted by heavy metallic connections, is not safe.

In the last decade a new type of robot has been analyzed, thanks to the research and the evolution of materials. These, called soft robots, are mobile machines constituted by flexible materials, such as fluid, elastomer or gel. They are designed taking hint from behaviour of animals such as octopus or worms. Without a rigid body these robots allow for flexibility, adaptability for performing tasks and promise a high degree of versatility and robustness, and are generally safer to interact with humans than robots of conventional design (Albu-Schäffer and Bicchi, 2016). For example they could change their size and shape to overcome an obstacle, moving through objects or on different surfaces, adapting themselves to different surroundings. When a soft-bodied robot collides with an object or with the environment, the impact is absorbed by their compliant structures, thus reducing damage (Salisbury et al., 1988).

Moreover, there is the knowledge given by the Swarm Robotics. This is a study inspired by social insects, such as ants, bees or wasps, with the aim to control a collective behaviour of a large group of simple robots to achieve a given task. Indeed, there is the idea that multiple simple robots can be more efficient to perform complex tasks than a single elaborated one. The researches were inspired by the insects, which can coordinate their activities, and then the behaviour of the entire group, through a communication between the individual elements. Hence, different types of robots have been developed, with the capacity to organise themselves in groups and working together for a specific purpose.

The idea of this project is born from the collaboration of these two robotic fields, namely the creation of a robot characterized by the features of a soft robot and the typical properties of robustness and scalability of a swarm of robots. Indeed, several soft-bodied robots are realized and designed to be controlled by a single processor or control, but, a different direction is chosen, applying the principle of collective intelligence to the idea of a flexible system, obtaining a configuration scalable and more robust from failures. Moreover, while some soft-bodied robots are of a modular design (Onal and Rus, 2012), the connections are usually linked irreversibly at the fabrication stage. A less explored alternative are soft robots that are modular and reconfigurable. The shape and the size of this *swarm* could be changed to cope with new and unknown situations to solve problems that would be otherwise impossible for a single robot. An example of application could be the exploration of new areas, characterised by different dimensions, in which a reconfigurable and deformable robot is useful to adapt to the new area.

A variety of mechanisms are suggested for actuation of the soft-bodied robot. Tensile actuators, variable in length, can be integrated into a soft structure to produce deformation (Brochu and Pei, 2010). A deformation can be produced in a soft structure by tensile actuators, variable in length or by pneumatic and hydraulic actuators which apply pressurised fluids to internal paths of the structure to deform the shape of the robot.

This project describes the assessments that brought to the realization of the Kilobot Soft Robot, a novel soft-bodied modular reconfigurable robotic system. Moreover, this dissertation describes the algorithms designed to control the motion of the Robot and the analysis of its movement properties. The modules are autonomous and are based on the openly available Kilobot platform, which is a very simple and cheap solution realized for swarm robotics applications (Rubenstein et al., 2012, 2014). In order to accomplish our aim, the Kilobots are inter-connected via reconfigurable, elastic links (springs), and are actuated using two vibration motors, the ones commonly found in mobile phones, making the realization of the Kilobot Soft Robot cheap. Moreover, they can communicate and measure their respective distances within a certain neighbourhood. Different types of spring are considered and tested in order to select the best in performance. A ring-shaped holding structure is designed to allow a mechanical attachment of the spring to the module, without limiting or affecting the movement of the module itself. The units and links assume a square lattice configuration, the default shape of the Kilobot Soft Robot. The dimension of the system can be manually reconfigured to suit different tasks. Each module is actuated to influence locally the configuration of the group. The aim is to control the movement and the shape distortion of the whole soft-bodied system through the combination of the motion of every Kilobot. Hence, the connected modules self-localise and move, while being constrained by the springs, using only local information. For this purpose distributed algorithms are designed to enable the Kilobot Soft Robot to follow a desired trajectory, to squeeze and extend its shape and to have a control on its motion independently from the number of modules that constitute the Robot. In order to test these properties in practice, several experiments are conducted with up to 49 modules constituting the system. To the best of our knowledge, this is the first demonstration of a fully autonomous soft-bodied modular reconfigurable robot of this size (in terms of number of modules).

The project, introduced above, was accomplished in the *Natural Robotics* research group at the Automatic Control and System Engineering (ACSE) department of the University of Sheffield. This dissertation discusses the project and is organised as follows. In the following Sections of this Chapter we introduce the methods followed to design the Kilobot Soft Robot and we discuss other studies that investigated modular and soft robotic solutions. In Chapter 2 we present the hardware description of the soft-bodied system and the implementation of the decentralised algorithms executed on each module. Following we report in Chapter 3 the experiments conducted to show and validate the robot's capabilities. In particular 3 properties are analyzed in the experiments: the performance of the Robot with increasing its size, up to 49 modules (Section 4.2), the ability of the Robot to follow a reference trajectory (Section 4.3) and modify its shape (Section 4.4). Finally, we conclude with the discussion of the outcomes of this study and the possible future works, in Chapter 5.

1.2 Aims and Objectives

The main objective of the project is to realize and control a cheap, reproducible, modular and reconfigurable soft-bodied system, which is built inter-connecting a group of autonomous modules. The first step of this study is to investigate how the single elements of the system with limited communication range and distance detection can self-localise. This means to investigate the possible ability of every module to define the positions of its neighbours. A first part of the algorithm is designed and tested to solve this localization problem. After becoming familiar with the platforms used, the Kilobots, a proper way to inter-connect them through elastic links (springs) has to be found. A 3D printed external structure is assembled with the module and designed to not influence it during its motion. The springs are manually realized from a plastic material. The final aim is to complete the distributed algorithm to control the movement and the shape of the Kilobot Soft Robot. The control is implemented on each module and is based on the mutual communication with the neighbours, on its local position in the group and on the constraints introduce on the movement by the springs. Finally, the results of several experiments are analysed and plotted in order to find particular features of the built Robot and to show its abilities.

Chapter 2

Literature Review

This study is born from the idea to obtain a modular soft-bodied system starting from a swarm of robots. Studies on soft robotics are conducted bringing to the realization of different examples of flexible robots, some characterised by a soft structure, a modifiable skeleton which can change its stiffness, others born from a composition of modules. In the literature, modular or reconfigurable robots exist, and we propose a soft-bodied system based on omnidirectional mobile robots (the Kilobots) joined via passive elastic links. The resulting lattice structure can move in a planar environment while adapting its shape and has the properties of modularity and reconfigurability. This approach takes inspiration from the principles of Swarm Robotics: the social forces in groups of animals are reproduced to create a group of elements that can move in a coordinated manner. The following Sections discuss the similar features between the Kilobot Soft Robot and other studies already investigated and describe the platform used, the Kilobot, designed for Swarm Robotics applications.

2.1 Related Work

To obtain robotic systems that are able to adapt to different environments and tasks, modular robotic systems have been widely studied. The final long-term objective is the realisation of a robot able to change its shape in an arbitrary manner on demand (Yim et al., 2007; Gilpin and Rus, 2010).

To obtain shape changing capabilities, system are designed to be composed by modules with a regular shape that can easily attach and detach composing more complex shapes. Furthermore, robotic modules have been developed in several different sizes, from micro to macro, in order to address different scenarios.

Most commonly, the movement of modules is restricted by the rigidity of the connecting links. A few platforms allow individual modules to exert forces into arbitrary directions, while being connected (Hirose et al., 1996; Baldassarre et al., 2007; Murray et al., 2013; Mathews et al., 2017).

Besides, changing the shape and the size, the modular robotic concept can be used in the soft robotics domain, obtaining deformable modular structures. A solution is proposed by Lee et al. (2016) and Wang and Ahn (2018), in which a modular system is constituted by soft robotic blocks connected by different types of rigid links. These systems are typically defined by modules with soft actuators, and reconfigurable links. For instance Kwok et al. (2014) use pneumatic systems to actuate silicone elements; the modules are connected through magnetic links.

Moreover, through a proper combination of the modules, it is possible to obtain a system with adaptable flexibility. For instance, this happens when modules are connected through links that can change their length, and hence change the shape and the flexibility of the structure. This is typically achieved by using active links, such as in (Hamlin and Sanderson, 1995), capable of changing their length in a controlled manner.

Yu et al. (2008) propose the use of both active and passive links: while the active links are controlled to expand or contract using motors, the passive links expand or contract according to external forces. Those links are used to connect cubic elements, creating a deformable 3-D structure.

Taking inspiration from the work of Yu et al. (2008), in this study we realise and investigate the Kilobot Soft Robot, a soft-bodied modular robot that is based on simple active robotic modules (omnidirectional mobile robots), connected through passive elastic links (springs). The result is a lattice structure that can move in a planar environment controlling its shape. This concept takes inspiration from the studies carried out under the multi-robot system works and swarm robotics, in which virtual potential fields are used for the coordination of the motion of the robots. Those virtual potential fields are designed to produce attractive and repulsive forces, as in (Gazi and Passino, 2004), whose combination leads to an aggregated behaviour of the swarm of robots. This approach takes inspiration from the social forces observed in groups of animals, as discussed by Reif and Wang (1999) and Reynolds (1987), and reproduced to create groups of robots that can move in a coordinated manner. Moreover, artificial potential fields are used to create geometric formations, as in (Leonard and Fiorelli, 2001). In this paper, we replace the artificial potential fields with physical elastic links, thus imposing real forces to constrain the motion of the robotic modules, defining a soft-bodied robot.

2.2 The Kilobot Platform

In order to realize the Soft Robot a defined number of Kilobots are inter-connected through springs. The Kilobot is a very cheap solution which provide all the main tools required to control a group of robots, such as sensors of communication and distance detection and a propelling system.

The Kilobot is characterised by three legs, which are only a support to the

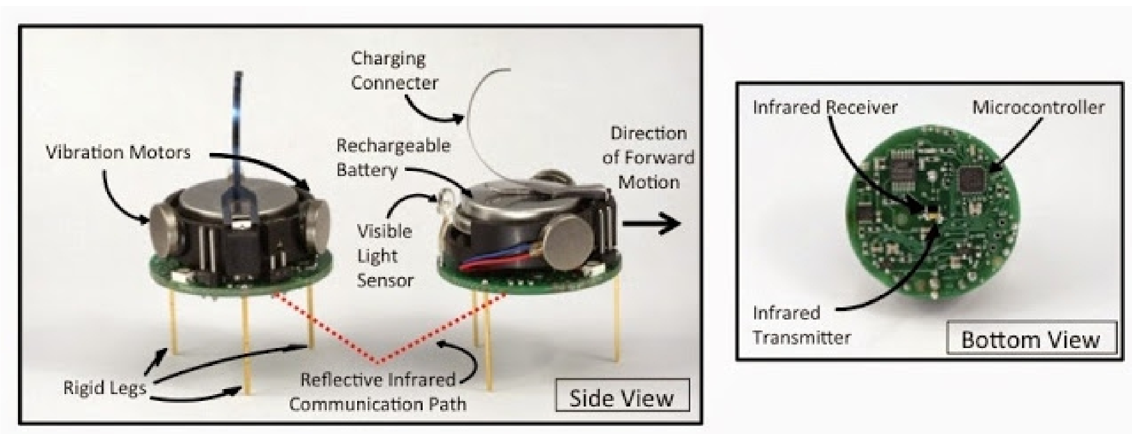


Figure 2.1: Different views of a Kilobot and its components.

body. The module has a round shape body of 3.3 cm radius and a height of 3.4 cm including the legs. All the components are showed in the Figure 2.1: a charging connector essential to charge the device battery, two vibration motors, one ambient light sensor, three rigid legs, one RGB LED, a lithium-ion rechargeable battery and infrared receiver and transmitter.

The motion is based on the slip-stick effect obtained through the vibration of the device, which means that the motion consist on the alternation between sticking and sliding. Depending on the battery level and the surface used, the Kilobot optimally moves at 1 cm/s and turn at 45 deg/s. The IR receiver and transmitter are located under the body and the signal is transmitted bouncing on the surface, in such a way all the neighbours can receive the message. The transmission range is at 15 cm on average and depends on the battery level and the surface used. The intensity of the IR signal is used to estimate the distance from the transmitting module. Each message is transmitted twice per second and can carry a maximum of 9 byte of data. The choice of the surface is important, it should be smooth and reflective to optimize both motion and communication. Each Kilobot is provided of a Atmega328 microprocessor, which runs at 8 Mhz and has 32 Kb of memory.

The simplicity of this platform allows to keep low the costs and, hence, to conduct

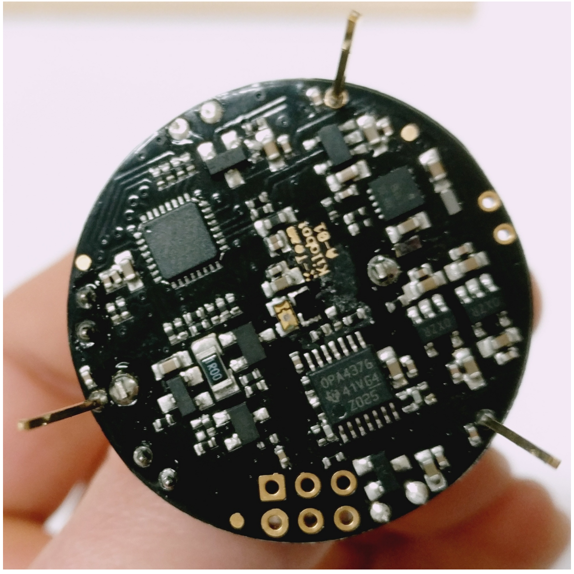


Figure 2.2: Different views of a Kilobot and its components.

experiments characterised by hundreds of units. However, the Kilobot is limited in performance due to this cheap hardware and is unique in features. The mobility is one of the most problematic aspects: the motion is inaccurate and noisy since it is based on vibration. The device lacks any form of odometry, and the only feedback of its position and movement can be provided by the neighbours. Hence, a good self-localisation based on the mutual communication with others is important. The transmission is not reliable, it depends on the ground surface, battery level and noise, since it is based on IR communication and collisions between the signals can corrupt the transmission of the message. Collisions usually happen when several devices transmit information at the same time in a small space. In this study the constraints on the movement applied by the springs, used to built the Robot, help in the motion control of the modules. The following Section describes how the Kilobot Soft Robot is realized, how the modules are disposed in a lattice configuration and how the elastic connections are attached to the device bodies.

Chapter 3

The Kilobot Soft Robot

In this chapter we explain the design and implementation of the Kilobot Soft Robot. Sections 3.1 and 3.2 present, respectively, the hardware solution and the algorithms for estimation and control.

3.1 Hardware platform

The modules of the Kilobot Soft Robot are arranged in a $r \times c$ square lattice configuration, as shown in the Figure 3.1, with r is the number of rows and c is the number of columns. The number of modules can be manually modified before each experiment, which means that the system is reconfigurable. Each module is a Kilobot, is able to turn left or right and move forward and is unable to rotate on the spot or move backward. Moreover, the unit is subjected to restrictions on the mobility due to the elastic links. These are implemented using helical springs between the modules. While the Kilobot turns, a force, proportional to rotation and elastic coefficient, is generated by the spring and applied to the device. The Kilobot has a range in which it can rotate and move freely until the limits in which the motion is hindered due to the extension of the spring.

As described previously, each module has a IR receiver and transmitter allowing the exchange of messages within a range of 15 cm. The length of the elastic connections is chosen in such a way the distance between the Kilobot and its neigh-

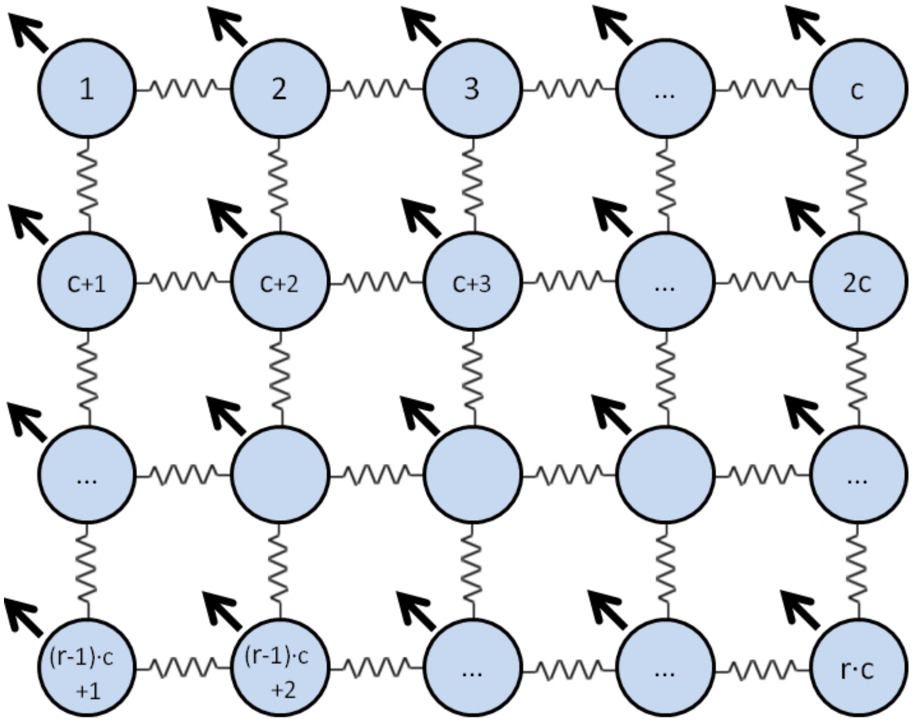


Figure 3.1: The Kilobot Soft Robot is constituted by Kilobot modules interconnected by springs, forming a $r \times c$ lattice configuration. Every Kilobot is ordered and identified by the unique number ID that is shown. The IDs can be either manually or automatically assigned. All the modules are oriented approximately in the motion direction, indicated by the arrows.

bours, during the motion, is always within that range, to be sure the transmission is not interrupted. However, due to hardware differences among the devices and communication noise, the transmission is not completely reliable.

The elastic links are implemented though helical springs, which were produced from a $500\,\mu$m acetate sheet, since the commercially available ones are too heavy. The sheet was sliced in $\sim 800\,\mu$m thick of filaments. These were winded around a tube of 1.4 cm diameter and then heated to deform it into the helical shape. The resulting springs, shown in Figure 3.3(a), have a length of $\sim 3.1cm$, a weight of $\sim 0.0425g$, and a spring coefficient of ~ 0.6.

To attach the springs to the Kilobot body, we designed a ring-shaped holding structure (Figure 3.2) and realized through to the 3D printer provided by Sheffield Robotics. The structure has a diameter of 4 cm and weighs 1.4 g. It is mounted,

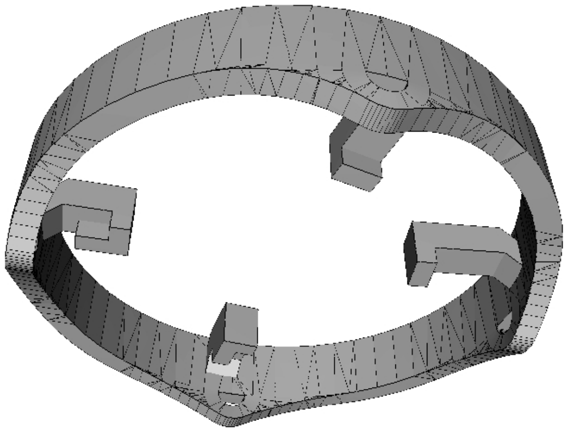
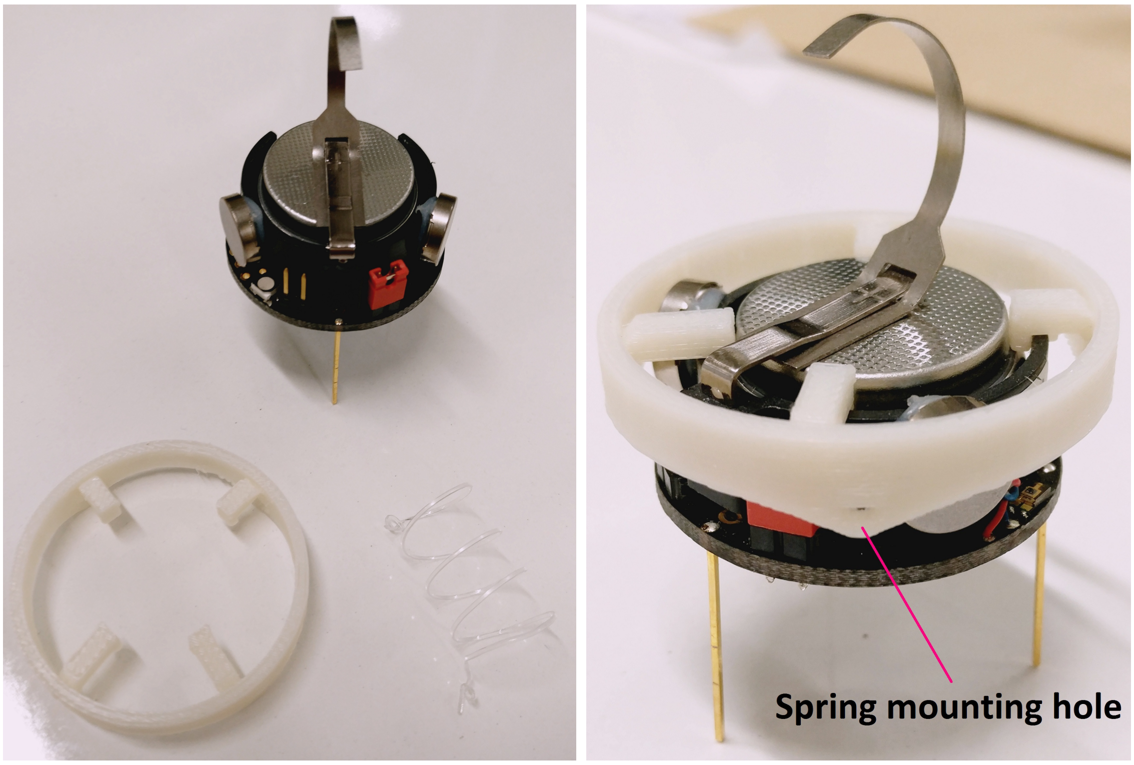


Figure 3.2: The external holding structure designed to allow the connections between the Kilobots and, hence, the realization of the Kilobot Soft Robot.

attached to the battery, on top of the module, which has a diameter of 3.3 cm and weighs 17.2 g. The structure is design to be well secured on the body and to avoid any interference with the vibration motors, communication system and to allow the Kilobot to display its RGB light-emitting diode (LED). As Figure 3.3 shows, the structure has four small holes equally spaced on the bottom of the ring. At the ends of each spring two small knots are realized allowing the user to lace the spring to one of the structure four holes. Hence, each Kilobot has a maximum of four neighbours attached to it through four springs positioned at $45°, 135°, 225°, 315°$ with respect to the forward motion direction.

During the experiments modules positions where recorded by a complex system (ARK) provided by Sheffield Robotics (described in Chapter 4). This is constituted by four cameras, which identify and record the Kilobots, and several IR transmitters in order to communicated with the Kilobots. In the design of the 3D printed holding structure not only the features needed for this study were considered. It has been design to facilitate the identification of the Kilobot and to improve the LED color detection from the cameras. Indeed, the localization and in particular the LEDs detection from the ARK system of a large swarm of Kilobots was an unsolved problem. The designed structure was employed to solve it and, hence, to facilitate the motion tracking by the cameras of a high number of Kilobots. In parallel with



(a) (b)

Figure 3.3: Overview of the module of the Kilobot Soft Robot. (a) Up to four helical springs can be connected to the Kilobot through a 3-D printed ring-shaped holding structure. (b) Kilobot equipped with the holding structure, which is assembled on the battery.

this project, hundreds of holding structures were printed (see Figure 3.4) and used in swarm robotics research activities (Talamali et al., 2019).

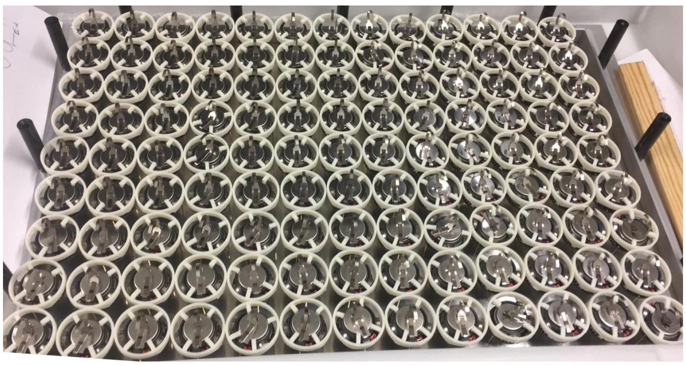


Figure 3.4: Several Kilobots improved by the designed 3D printed external structure. These were used in other projects and studies related to swarm robotics (no interest in springs connections).

3.2 Algorithms

Each module in the Kilobot Soft Robot, mechanically coupled with its neighbours is still fully autonomous. Robustness and modularity are some of the features implemented through decentralized algorithms executed by every member of the group. The modules are thus fully inter-changeable. The Kilobot Soft Robot can be arbitrarily reconfigured without any change in the code. The size of the Robot can be increased or decreased without affecting the global performance thanks to the modular and scalable design. In fact, the general algorithm, written in C programming language, allows every Kilobot to determine its own position (among the nine possible, see Figure 3.8 and Section 3.2.1) in the lattice starting from the knowledge of its own ID number and the numbers of modules that constitute the Soft Robot. The method to obtain this information is generalized and does not depend on the system size. The scalability was tested up to 49 modules constituting the Soft Robot (see Figure 3.5).

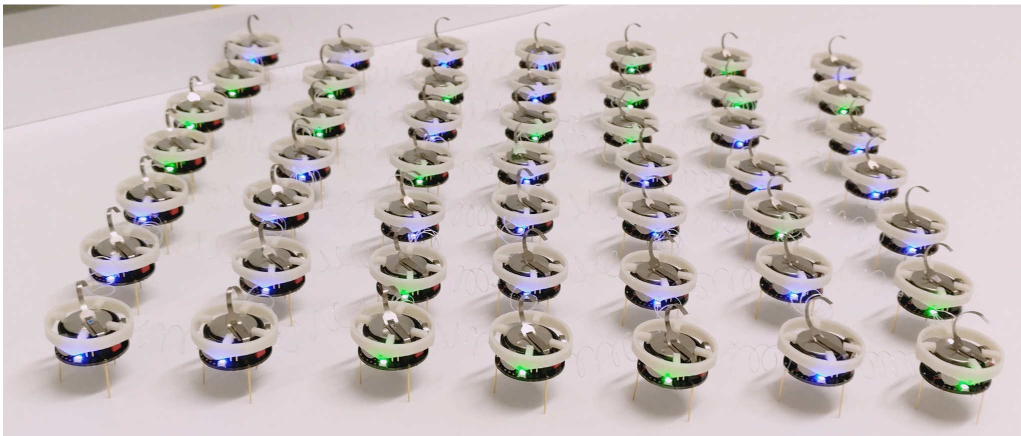


Figure 3.5: A 7x7 Kilobot Soft Robot composed of 49 modules connected by 84 light-weight springs. The LEDs show the battery level. The battery is considered discharged if the light is yellow or red.

Before deciding if and where to move, a module estimates its location relative to its neighbours. Kilobots are devoid of sensors able to detect the environment, so they rely on the information provided by the others units. Each module self-localise using the distances communicated by the neighbours. Afterwards the module determines a motion command considering both distances detected and angles computed. The following Sections discuss the *localisation* and *motion control* algorithms.

3.2.1 Localisation

The aim is to enable every Kilobot to determine its neighbours positions. In this manner, every module can correct its movements according to the localised modules around it.

The *localisation* algorithm is based on an efficient sharing and storage of the detected distances, and a quick computation of the angles, once defined a triangular shape among the modules. Two algorithms were designed: one starts from the assumption that the modules are already identified by a unique ID and disposed tidily to form the Soft Robot, the other starts only from the consideration that the Kilobots are identified by randomly assigned IDs. The following Sections discuss both of these algorithms, the first is meticulously described since it is the one used to control the Kilobot Soft Robot for the experiments, and the second is only quickly illustrated.

IDs tidily assigned

The aim is the local estimation of the module position relative to its neighbours or, in other words, the localisation of the module neighbours relative to a local reference system fixed on the module. This task is challenging: we know the Kilobot can estimate the distance, measuring the signal intensity, from the module is transmitting, this does not means knowing its position. To have a localisation the computation of the angles between the two Kilobots and a third reference one are needed. The springs simplify the solution to the problem, as the modules of the Kilobot Soft Robot can not be positioned arbitrarily (see Figure 3.1). The Kilobots are inter-connected by springs forming quadrilateral figures. Hence, a swap of two modules is mechanically not possible, and hence an ambiguity in the localisation is avoided. Moreover, we assume that, building the Kilobot Soft Robot, each module is assigned a unique identifier (ID) and all of them are sequentially disposed to compose a lattice of r rows and c columns (see Figure 3.1). The advantage is that each element, knowing its own ID i, can determine its own position in the lattice (e.g. on a corner or on an edge, see Figure 3.8) and all the neighbours IDs before starting the communication. For example, the modules IDs in the neighbourhood are $i-1$ (for $i \mod c \neq 1$), $i+1$ (for $i \mod c \neq 0$), $i-c$ (for $i > c$), and $i+c$ (for $i \leq rc-c$), where i is a general ID referred to a module in the lattice.

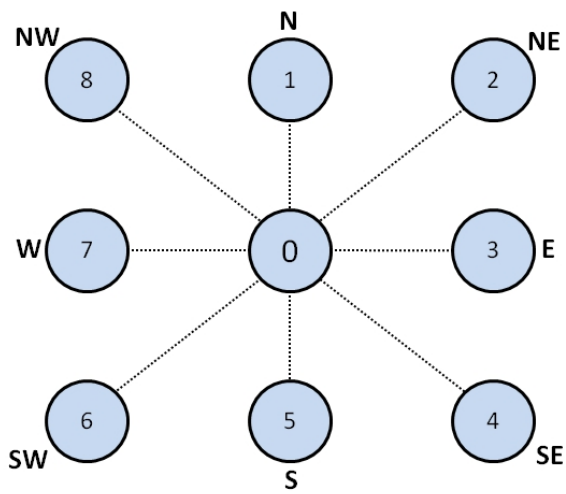


Figure 3.6: Each module has up to eight neighbours. This is a frame of the system shown in Figure 3.1. Labels show the intuitive method used in the code to indicate a relative distance: "NW" refers to North-West.

The number of neighbours can vary depending on the position of the module in the lattice and is maximum eight, as shown in Figure 3.6. Each Kilobot continuously sends and receives messages, so estimating distances, during the motion. The motion control (Section 3.2.2) adapts itself to the new values computed.

The transmission is ready twice per second, while the reception is much quicker. A procedure composing the message is defined to optimize the communication (see Figure 3.7). Each module sends always its own ID to allow the receiving modules to associate the detected distance with the relative neighbour. A filtering is applied on the messages read, since the Kilobot receives all the signals in the permitted communication range, not only from its neighbours (see Section 2.2). Hence, if the first data byte of the message received carries a neighbours ID (maximum eight possibilities), the message is read, otherwise it is ignored. Every Kilobot is interested in saving the distances estimated by every its neighbour, so everyone has to send "outside" the eight (or less, e.g module on a corner) distances detected. Note that it is not necessary to send the respective ID of the robot from which the signal is received and the distance is estimated. A rule is followed to fill the message: this can contain 9 bytes of data, since one is occupied by the own ID (first byte), the others 8 tidily carry all the distance estimates (one byte each, see Figure 3.7) for the eight neighbouring modules (where available). The receiving Kilobot already knows, from the byte position in the message, which is the neighbour of the transmitter a distance refers to.

For example, if the Kilobot is positioned on the lattice corner, only three distances are estimated and, then, transmitted. It is not necessary to send the relative IDs together with the distances since every module fills the data bytes with a predefined order. For example, when a message is received, the module knows that the 2nd data byte carries the distance between the transmitting module (ID in the 1st data byte) and its *Nord* neighbour. In this manner, the estimated distances with all the neighbours are shared through a single nine bytes message (twice per second, see Section 2.2).

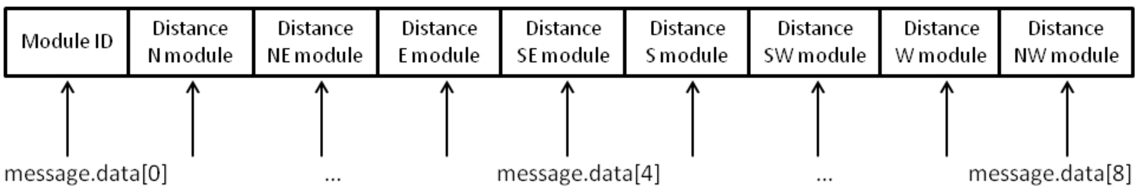


Figure 3.7: Structure of the message exchanged by the modules to share estimated distances. In order to optimize and speed the algorithm a message protocol is defined.

Initially, the module has no estimate of any distance. When the communication starts, it stores all the distances directly measured and also the ones carried by the message in a 9×9 matrix. In this manner, each module can save and easily have access to all the possible distances between the modules in the neighbourhood. For example, in the first row and column of the matrix only the distances between the

Kilobot and its neighbours are stored since the own ID is the one with position "0" (the neighbours in the Nord has position 1, as shown in the Figure 3.6). Note that the matrix is typically not completely filled, since the distances shared are only the ones estimated among the eight neighbours, some Kilobots can be out of the communication range, or the module is located on the boundary of the lattice. Hence, for example, the position in the matrix that carries the distance between the N and SE neighbours will be empty. The matrix is reset periodically (every 2 s) to clear outdated information, and hence to avoid that a loss of communication brings a wrong value. Note that the estimates for pairs (i, j) and (j, i) usually differ for $i \neq j$, despite they refer to the same distance, since each Kilobot is unique in terms of hardware and battery level and the intensity of the transmitted IR signal (through which the distance is measured from a neighbour) could vary. The average is computed where both values are available and then substitutes for them in the matrix (lines 11-14 algorithm 1).

The final step of the *localisation* algorithm is the computation of the angles. Two modules are chosen among the neighbours and the three distances involved, saved in the matrix, are used to compute the three relative angles following the *law of cosines*. Not all the angles are in the lattice are computed but only the ones used in the *motion control* algorithm (see Figures 3.8 and 3.9). Making an example to understand better the process: a triangular shape could be defined among the adjacent Kilobots in positions P_1, P_2 and P_4 (see Figure 3.8), the last two are physically connected with the first and a virtual connection can be imagined between the Kilobots in positions P_2 and P_4. These three virtual and physical links correspond to three distances known by the three modules involved (see Figure 3.9). The three relative angles, inside the defined virtual triangle, can be potentially computed by each one of these three Kilobot thanks to the *law of cosines*. However, this computation is executed only by the one in position P_1, since this Kilobot uses in the *motion control* algorithm (see Section 2) one of these angles, in particular the one adjacent showed by a green arc in Figure 3.8 and 3.9. Similarly, every Kilobot in the lattice computes one or two (only the modules in position P_5) adjacent angles starting from three distances. For example, the computation executed on the Kilobot in position P_3 uses the distances "$P_2 - P_3$", "$P_6 - P_3$" and "$P_2 - P_6$" (the adjacent modules in Figure).

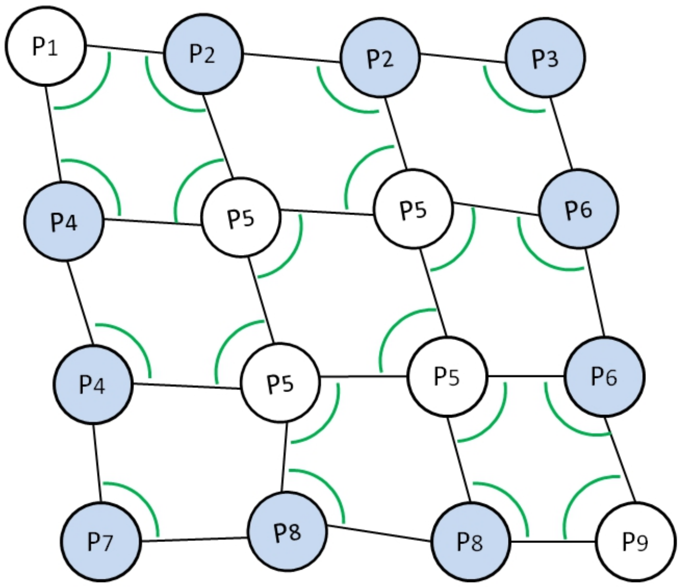


Figure 3.8: General case of the Kilobot Soft Robot in which the angles computed by the relative modules are illustrated. Note that the labels (P_1 to P_9) enumerate the nine different situations a module can reside in. For example, a module of label P_5 is in the internal part of the lattice, and hence has 8 neighbours and two angles to compute. The module with label P_3 is in the top-right corner of the configuration. The white modules compute the angle(s) parallel to the motion direction, while the blue modules compute the angle perpendicular to it.

In Figure 3.8 are shown the relevant angles used and computed since they are, in general, sufficient to determine if the shape is deformed or not and to design a motion control for the Kilobot Soft Robot. Each module estimates only one or two angles: in Figure 3.8, the white modules compute the angle(s) parallel to the motion direction, while the blue modules computed the angle perpendicular to it. The numbers in the Figure 3.8 are not the IDs, but they show the nine different cases that are considered in the algorithm. These are different in number and disposition of the neighbours. For example, the interior modules (labelled with P5) have all their eight neighbours and are the only ones that compute two angles. If this module has ID i, it will calculate the angles between neighbours $i - c$ and $i - 1$, as well as between neighbours $i + c$ and $i + 1$. The reference value is 90 degrees since the

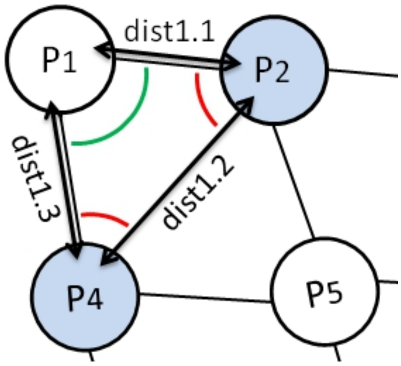


Figure 3.9: An enlargement of the Figure 3.8, in which the triangular shape defined to compute the angles is underlined. The three distances $dist_{i.1}$,$dist_{i.2}$ and $dist_{i.3}$, where $i = 1$, are necessary to potentially computed the three angles. However, the *motion control* algorithm uses only one angles per each module (except the ones in position P_5), so only the interested angle (the green one) is computed.

lattice is the default configuration of the Kilobot Soft Robot. The angles variables are reset together with the matrix of distances every two seconds to avoid a storage of a wrong value if the communication is lost. While the distance estimates are continuously updated, the relative positions in terms of angles are computed at the beginning of every cycle of the motion control algorithm.

The illustrated localisation method is efficient when we take into account the physical links that exist among the modules constraining them into a lattice configuration. Indeed, in general, the information provided by angles and distances estimated by two modules are sufficient to uniquely identify a third one. Given the distances measured by two modules to localise a third one, the Figure 3.10 shows that there exist two possible relative positions for the third module. These two configurations are equal in terms of angles and distances, this means that three Kilobots are necessary to uniquely localise a neighbour. This ambiguity due to partial information is solved by the presence of physical links and by the lattice topology of the Robot (see Figure 3.1). The modules are free to move, a distortion of the angles is allowed thanks to the elastic links, but they can not be involved in a swap of the positions since the magnitude of angular distortions is limited by the presence of the physical connections. Hence, knowing the lattice configuration of the system with all reference angles at ideally 90 degrees, it is possible to unambiguously identify

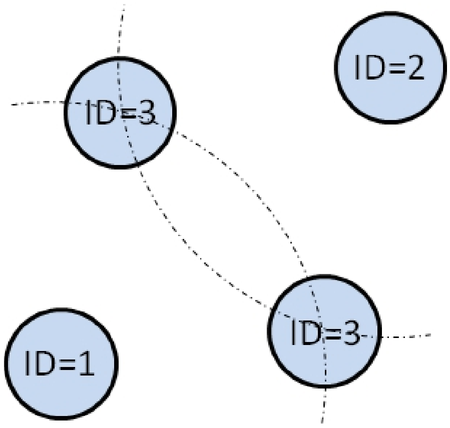


Figure 3.10: Given the distances measured by two elements (ID 1, ID 2) to localise a third one (ID 3), there exist two possible relative positions for the third module. This ambiguity can be removed by taking into account the lattice configuration assumed by the Soft Robot. The modules can not be involved in a swap of the positions since they are mechanically inter-connected and the magnitude of angular distortions is limited by the presence of these links.

and localise the neighbour.

Once the *localisation* algorithm is verified, the *motion control* algorithm is designed. This uses some of the angles, already shown in Figure 3.8, and estimated distances to implement a motion control of the Kilobot Soft Robot. The next Section describes the structure of the algorithm and which data are used to implement the control on the module.

Algorithm 1 Localisation algorithm for module at position P_i with $i \in \{1, 2, \ldots, 9\}$ as specified in Figure 3.8

```
procedure LOCALIZATION
    set the neighbours IDs in a 9-element array V
    Transmit own ID to the other Kilobots
    while true do
        if message transmitted successfully then
            Transmit own ID and saved neighbours distances in a single message
        if message received then
            if received ID is one of the ID in V then    ▷ filtering the distances detected
                save in the matrix of distances A the distance detected
                save in the matrix A distances carried in the message
        if A[j][k]! = 0 AND A[k][j]! = 0 then    ▷ A is a symmetric matrix
            compute the average value
        else if A[k][j] == 0 AND A[j][k]! = 0 then
            A[k][j] = A[j][k]
        if dist_{i.1}! = 0 AND dist_{i.2}! = 0 AND dist_{i.3}! = 0 then    ▷ executed twice for i = 5
            compute the interested angle
        motion control algorithm
        if time > n * 32 then    ▷ n seconds, n = 2
            set matrix of distances A = 0
            set angles = 0
```

IDs randomly assigned

As previously the aim is the local estimation of the module position relative to its neighbours. However, now we do not assume a sequential organized disposition of the modules uniquely identified by an ID. We suppose the IDs are randomly assigned among the modules that compose the Kilobot Soft Robot. This algorithm was not used because introduces in the localisation process some problems and complexities, which are solved and simplified by the predefined and ordered ID assignment, discussed in the previous Section. However, it is debated in this Section in order to show the main problems, to make a comparison with the previous algorithm and because it could be a good starting point to design a new localisation method. Indeed, the IDs can not always be tidily and automatically assigned to the Kilobots in the lattice, it may be necessary a manual assignment. Differently, a random and automatic assignment is usually possible. The concepts here described take inspiration from the previous algorithm and, hence, they present some similarities with the ones already discussed.

In the algorithm each Kilobot continuously sends and receives messages, estimating the distances, from the neighbours, whose number depends on the position of the module in the lattice. The first main difference with the previous algorithm is the protocol message transmission. One data byte, out of nine, is dedicated to the Kilobot's own ID, which is continuously transmitted in such a way its neighbours can keep updated the respective distance. The other 8 data bytes of the message carry both distances detected and relative Kilobot IDs associated to those distances, since now there is not a predefined neighbourhood and so a filtering of the messages. Every module estimates distances from all the other robot around it near enough to support the IR communication.

Hence, the maximum numbers of neighbours each Kilobot detects and identifies depends on how many different IDs and relative distances it is enable to store. Increasing the matrix size in which these data are saved, the dimension of the neighbourhood increases. To compare the procedure with the previous one we consider a nine by nine elements matrix, in which the first row and column are filled with the distances between the robot its self and its neighbours.

Since it is not possible to know the modules IDs without communicating them, half of the message is dedicated to the transmission of these data. For example, the data bytes 1, 3, 5, 7 carry the Kilobot ID numbers from which the distance, carried in the data bytes 2, 4, 6, 8 is measured. Therefore, this method allow the transmission of half the information transmitted previously for each message.

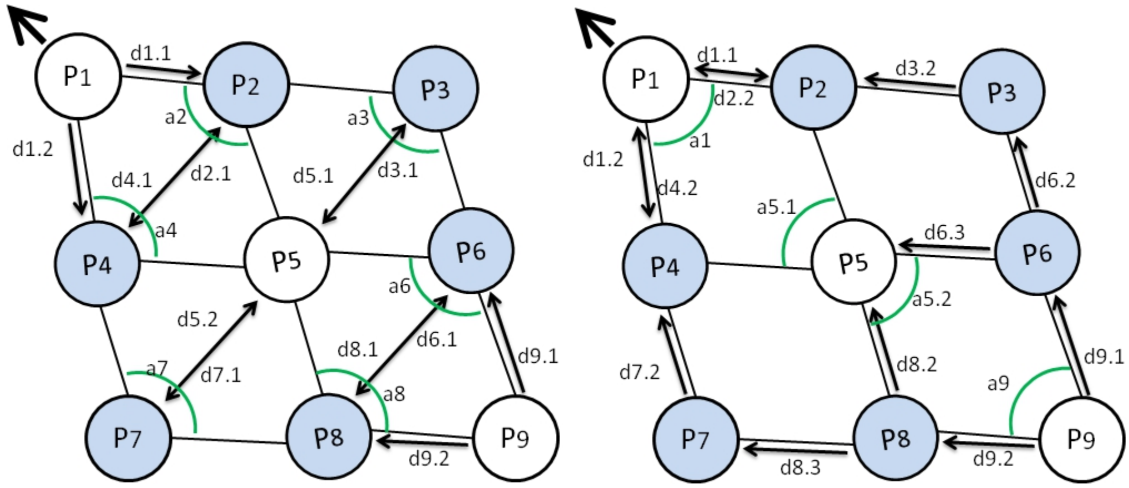
Moreover other complications shall be added: the matrix indexes are unknown and have to be updated and filled at the beginning during the communication, excluding the detected ones further away, since the Kilobot does not know a priory its neighbours IDs. The ambiguity problem, in the localisation among three elements, can not be solved thanks to the constraints imposed by the lattice shape, since the neighbour position is not predefined and, when identified by the robot, there is uncertainty on its localisation. Three modules are needed to unambiguously localise the unknown one complicating the computation.

Hence, in conclusion, the assumption of a Soft Robot composed by modules tidily disposed and identified is pursued, considered the several advantages this leads to the control design. Moreover, these simplifications in the algorithm are beneficial in terms of computational costs and memory occupied, given the low hardware performances of the Kilobot.

3.2.2 Motion Control

The purpose of the motion control algorithm is to move the Kilobot Soft Robot into a desired direction while maintaining its default shape, the lattice configuration shown in Figure 3.1. Hence, through this algorithm the mobility and shape control is implemented. In this Section we discuss about the method designed to make the Soft Robot keeping constant its default shape during the motion. To obtain a deformation of the Robot the same algorithm modified with different parameters is used, see the following Section. A 3×3 Robot was considered to design the control, since it is the minimal figure that contains all the different position a module can assume, as shown in Figure 3.8. The control of this system configuration can be extended to all the different sizes of the Kilobot Soft Robot. For example a 2×2 Robot can be seen as a 3×3 one deprived of all the Kilobots excepted the ones at the corners of the lattice. This is the simplest configuration with four distances estimated and four angles computed (one by each module).

The Algorithm 2 present the *motion control* algorithm, which is executed periodically by every module of the Robot. The algorithm is briefly illustrated in the diagram in Figure 3.12. To describe the methods followed we refers to a 3×3 system shown in Figure 3.11, in which the angles and the distances used to implement this control are illustrated. Every module estimates one angle, which is generally the one directed internally to the lattice, except for the modules in position P5, which compute the two angles parallel to the movement direction. Moreover, at least two



(a) (b)

Figure 3.11: Modules use their localisation estimates to determine if they are in the correct relative position to obtain the target robot shape or if there is a local deformation to be corrected. (a) and (b) show the set of distances and angles considered for, respectively, the first and second deformation assessments described in Algorithm 2 and relative Section. White and blue colours differentiate modules depending on their control motion algorithm as detailed in the text.

distances are used to complete the control and to prevent computation errors or loss of communication. Side and diagonal distances are considered: the firsts are oriented perpendicularly to the movement direction, the seconds are usually the ones in front of every module. The diagonal distances are important to change the shape from an extended to a squeezed lattice configuration. The side distances are used to avoid collisions between the Kilobots and to implement the last security control to check the motion of the module. Substantially, the algorithm consists in actuating every Kilobot in order to keep these angles and distances within a defined range. For example, the angles are maintained around the 90 degrees.

In the algorithm 2 each module desired direction is denoted by TARGET, which can assume three possible direction values (forward, left, right). By default, the target direction is forward, which means, if the Kilobot is in the correct position, i.e. all the conditions on distances and angles are satisfied, it will move forward. Moreover, a specific target direction can be provided (lines 3-4). When the Kilobot Soft Robot has the task to follow a trajectory, the module target direction is dy-

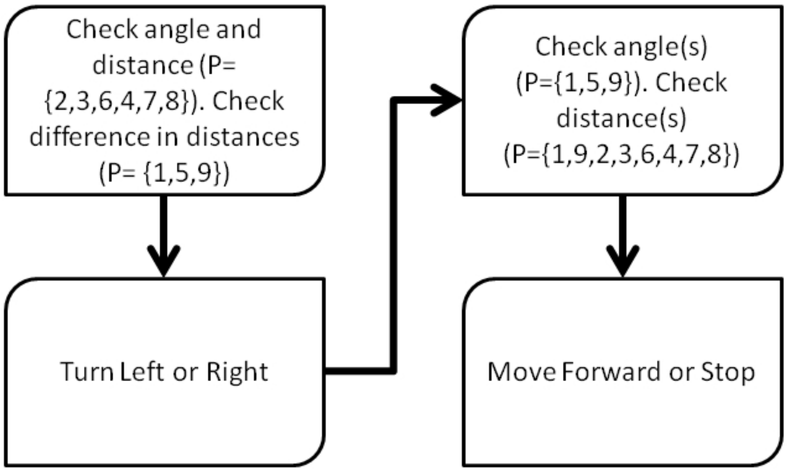


Figure 3.12: The diagram shows the two main steps of movement followed by every Kilobot. Note that the left part of the diagram refers to Figure 3.11a, while the right part to Figure 3.11b

namically variable during the motion in order to follow the specific path. This is discussed in the next chapter, which illustrates the experimental setups.

Each module decides to move forward, stop, turn right or left checking the positions of the neighbours and depending on where is positioned in the lattice. There exist nine possible positions, $P_1, P_2, \ldots, P_9$, shown in Figure 3.11. Considering the positions denoted by the index $i \in \{1, 2, \ldots, 9\}$, in the algorithm, a general module position is indicated by P_i. The estimated distances and angles used are all defined in Figure 3.11. While the distance estimates are obtained and updated in parallel to the motion control algorithm, the angle estimates are obtained at the time of use, in such a way to compute them with the most recent distance estimates. During the motion, a failure in the computation of the angle can happen, because at least one of the three distances needed in the computation is lost during the communication or has not been estimated yet. If at the time of executing the motion control algorithm a distance or angle estimate is not available, all expressions that make use of them are evaluated as false. The control implemented varies according to the position of the Kilobot in the lattice and it is the same for some groups of modules. For this reason we divide the modules into two groups, shown with white and blue disks in Figure 3.11. The white modules comprise those in the frontal (P_1), interior (P_5) and rear (P_9) positions. The blue modules are the remaining ones ($P_2, P_3, P_4, P_6, P_7, P_8$). The white modules execute lines 7–17 of the algorithm, whereas the blue modules execute lines 18–30 of the algorithm.

The motion command each Kilobot can perform divides the algorithm in two main parts: firstly the conditions on angles or distances are checked allowing the module to decide turning right or left, secondly, the module decides to move forward or stop after checking other angles or distances. Hence, every Kilobot motion is characterised by a rotation (if a position correction is required) and then by a straight motion or arrest (if the position is correct or not). Looking at the Figure 3.11a, the white modules turn right or left according to the distance estimates, while the blue modules turn after checking the relative angle and diagonal distance. Subsequently, looking at the Figure 3.11b, the white modules move forward or stop according to values of the angles, while the blue modules move after checking the front distance(s). Note that angles and distances used by a certain Kilobot are respectively indicated as green semicircles and outgoing arrows around that Kilobot.

In other words, following the algorithm 2, each module performs two deformation assessments, by analysing the spatial configuration of modules in its neighbourhood and checking if the module is in the desired relative position. During the first deformation assessment, the module may probe any of the distances and angles shown in Figure 3.11a, while during the second deformation assessment, the module may probe any of the distances and angles shown in Figure 3.11b. If a deformation is detected, at least one condition on the module position is not satisfied, during the first assessment (lines 8–11 for white modules, or lines 19–26, otherwise), the module executes a corrective move, by turning either to the LEFT or RIGHT (lines 9 and 11 for white modules, or lines 24 and 26, otherwise). If no deformation is detected, no corrective move is performed. The second assessment (lines 12–17 for the white modules, or lines 27–30, otherwise) determines whether the module may advance further, by executing a movement in the desired direction, or needs to wait for other modules to catch up. In the latter case, the module stops moving.

Some examples to understand better the algorithm: the Kilobot in position P3

Algorithm 2 Motion control algorithm for module at position P_i with $i \in \{1, 2, \ldots, 9\}$ as specified in Figure 3.8

procedure MOTION
  Let $a_i, d_{i.1}, d_{i.2}, d_{i.3}$ be defined as in Figure 3.11
  if module has `SPECIFIC_TARGET_DIRECTION` **then**
    `TARGET` = `SPECIFIC_TARGET_DIRECTION`
  else
    `TARGET` = forward
  if $i \in \{1, 5, 9\}$ **then** ▷ module either in frontal (P_1), rear (P_9), or central (P_5) position
    if $(d_{i.1} - d_{i.2}) > 1\,\mathrm{cm}$ **then** ▷ 1st deformation assessment
      turn right for 500 ms
    else if $(d_{i.2} - d_{i.1}) > 1\,\mathrm{cm}$ **then** ▷ 1st deformation assessment
      turn left for 500 ms
    if ($i == 1$ AND $a_1 < 85°$ AND $d_{1.1} > 7\,\mathrm{cm}$ AND $d_{1.2} > 7\,\mathrm{cm}$) OR
      ($i == 9$ AND $a_9 > 95°$ AND $d_{9.1} < 7\,\mathrm{cm}$ AND $d_{9.2} < 7\,\mathrm{cm}$) OR
      ($i == 5$ AND ($a_{5.1} > 95°$ OR $a_{5.2} < 85°$)) **then** ▷ 2nd deformation assessment
      stop motion for 500 ms
    else
      move towards `TARGET` for 500 ms
  if $i \in \{2, 3, 4, 6, 7, 8\}$ **then** ▷ module either on right robot side (P_2, P_3, P_6) or on left robot side (P_4, P_7, P_8)
    if $i \in \{2, 3, 6\}$ **then** ▷ module on right robot side (P_2, P_3, P_6)
      `ROBOT_CENTRE` = left
    else ▷ module on left robot side (P_4, P_7, P_8)
      `ROBOT_CENTRE` = right
    if $a_i < 85°$ OR $d_{i.1} > 11\,\mathrm{cm}$ **then** ▷ 1st deformation assessment
      turn towards `ROBOT_CENTRE` for 500 ms
    else if $a_i > 95°$ OR $d_{i.1} < 10\,\mathrm{cm}$ **then** ▷ 1st deformation assessment
      turn away from `ROBOT_CENTRE` for 500 ms
    if $d_{i.2} < 7\,\mathrm{cm}$ OR $d_{i.3} < 7\,\mathrm{cm}$ **then** ▷ 2nd deformation assessment, modules at $P_i \in \{P_2, P_3, P_4, P_7\}$ have a single distance ($d_{i.2}$)
      stop motion for 500 ms
    else
      move towards `TARGET` for 500 ms

during the motion is commanded to keep a lattice and extended shape, so aiming at a 90 degrees angle and a minimal maintained side/diagonal distance. The Kilobot will turn right if the angle is higher, with a tolerance, than 90 degrees or if the diagonal distance is too small, it will turn left otherwise (Figure 3.13a). Similarly, the Kilobot will stop if the distance detected from the front neighbour is to small, it will move forward otherwise (Figure 3.13b).

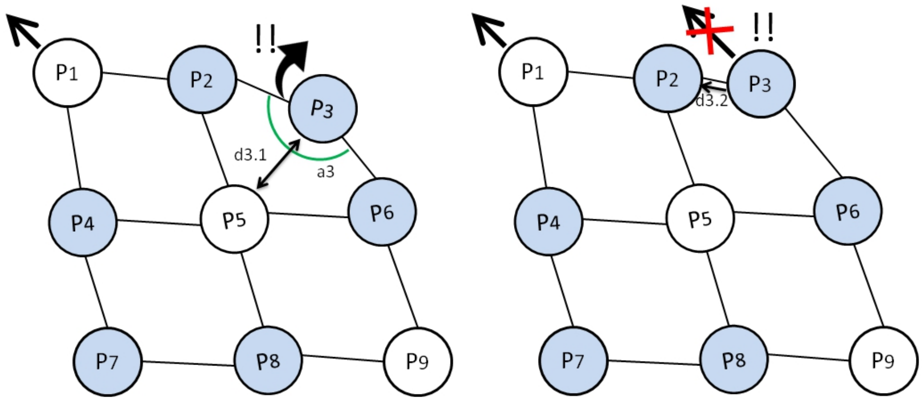


Figure 3.13: Example of motion control applied to the module in position P_3 (right corner of the lattice). (a) If the angle a_3 is higher than 95 degrees or the diagonal distance $d_{3.1}$ is lower than 10 cm, the Kilobot will turn right. (b) If the side distance $d_{3.2}$ is lower than 7 cm, the Kilobot will stop.

3.2.3 Deformation control

Due to the elasticity of the links between the modules, the Kilobot Soft Robot can modify its shape. This is particularly useful when the robot has to move through complex environments comprising small passages. By default, the motion control algorithm (Algorithm 2) assumes that the robot's base structure (a square lattice) is to be maintained. The algorithm can however be modified to realise a different target shape. Experiments were conducted to test the deformability performance of the Robot. The main structure of the algorithm implemented on the Kilobots does not change, it is extracted from the previous one altering the adjustment ranges of distances and angles. The Kilobot Soft Robot gains the ability to alternately contract and extend, as shown in the experiments described in the following chapter.

The *deformation* algorithm is achieved by changing the reference angles, from the default value of $90°$, using an offset α. Looking again at the Figure 3.11, the modules represented using white disks would use $90° - \alpha$ as the reference angle, while the modules represented using blue disk would use $90° + \alpha$ as the reference angle. For the deformation experiments, reported in the next chapter, a value of $\alpha = 50$ is chosen. Similarly, different reference values and conditions are considered on the distances. The blue module would move to minimize the respective diagonal estimated distance, while the reference range previously considered for the side distances would be increased in order to obtain an elongated shape supporting the squeezing of the Soft Robot.

Chapter 4

Experiments

This chapter illustrates the setups and results of the experiments conducted to evaluate the performances of the Kilobot Soft Robot. Three different types of experiments were conducted relative to the three different possible configurations of the system and its capabilities. During each trial, data on Kilobots positions and IDs are recorded and stored, subsequently these values are analysed and plotted. The first experiment, described in Section 4.2, shows the feature of the Kilobot Soft Robot to move straight forward without an external feedback (open-loop control) on the direction kept by the system. We will see how the performance increases with the increasing dimension of the Soft Robot size (number of modules). Characteristics as the shape distortion, speed or the deviation from the target will be measured as function of the Robot size. The second experiment, described Section 4.3, evaluates the ability of the Kilobot Soft Robot to follow a predefined reference trajectory. In this case the system is guided by an external feedback on its direction (closed-loop control). This feedback is provided by the ARK system, a technology made available by Sheffield Robotics (see Figure 4.2) explained in the following Section. The third and last experiment, described in Section 4.4, focuses on the ability of the Soft Robot to deform its body while following a predefined path. In particular, the system moves following a straight direction, first shrinking and then expanding its shape. All the experiments were conducted in the same arena, using the same system to record the data and under similar experimental conditions, with the tolerance of the human error. The common aspects of the experimental setup are described in the following Section, while the specific ones, as the starting and finishing Robot configuration for each run, are described at the beginning of each respective experiment Section. Videos of all the experimental trials are available at http://naturalrobotics.group.shef.ac.uk/supp/2019-002/ on the Natural Robotics Lab YouTube channel.

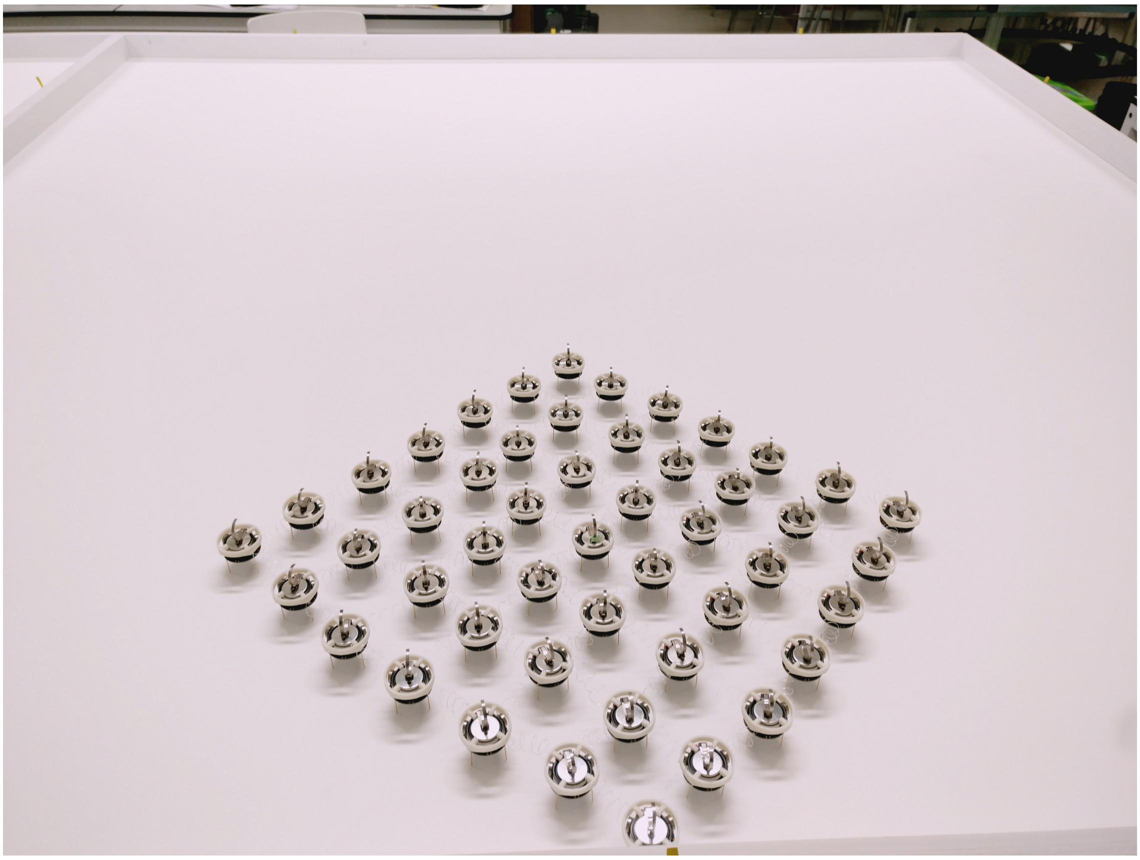


Figure 4.1: A 7×7 Kilobot Soft Robot at the beginning of a trial during the *straight motion experiments*. The modules are inter-connected by transparent springs, difficult to see from the photo. At the beginning the springs are at rest, hence the distance between Kilobot is $\sim$7 cm, measured from centre to centre.

4.1 Experimental base setup

The experiments were conducted on a flat, bounded $200\,\text{cm} \times 200\,\text{cm}$ arena made of whiteboard Perspex© Frost matt acrylic material (Moonlight White S2 1T41). All the trial are recorded trough four overhead cameras, the four images generated are matched in a single one. The system implements a real-time position tracking of each Kilobot seen under the cameras, distinguishing it from the neighbours through its ID transmitted to the overhead system. Moreover, it is possible to communicate via infrared with all the Kilobots that run in the arena in such a way to implement an external feedback. For example in two of the experiments, Sections 4.3 and 4.4, a feedback on the position with respect to a reference trajectory is provided to one of the Robot's modules. A system with these properties is realised using the Augmented Reality for Kilobots (ARK) technology (Reina et al., 2017).

ARK makes it possible to have virtual sensors on Kilobots, which means it is possible to record, track or communicate with the robots that run in the arena. The ARK system is constituted by a high number of overhead infrared LED transmitters to send in real-time addressed messages to each *augmented* Kilobot. This mechanism is used to enhance the Kilobot's capabilities, allowing the robot to follow a predefined trajectory. Substantially, a particular message is sent to every module according to their position with respect of a virtual reference path on the arena. ARK is openly available and has been proven to support swarms comprising hundreds of robots. In all experiments, ARK is used to record the positions of the modules for post-analysis.

In general, Kilobots mobility and IR communication are sensitive to the level of the battery, hence, to avoid a corruption of the experiments results all the runs are conducted after charging the robots, until the blue or green battery level displayed by the RGB LED. Moreover, every Kilobot is manually calibrated to improve the motion quality that is very noisy due to the vibration motors. At the beginning of each trial, the modules of the Kilobot Soft Robot are positioned such that the connecting springs are at rest (see Figure 4.1), hence, the centre-to-centre distance between them is $\sim 7\,\text{cm}$.

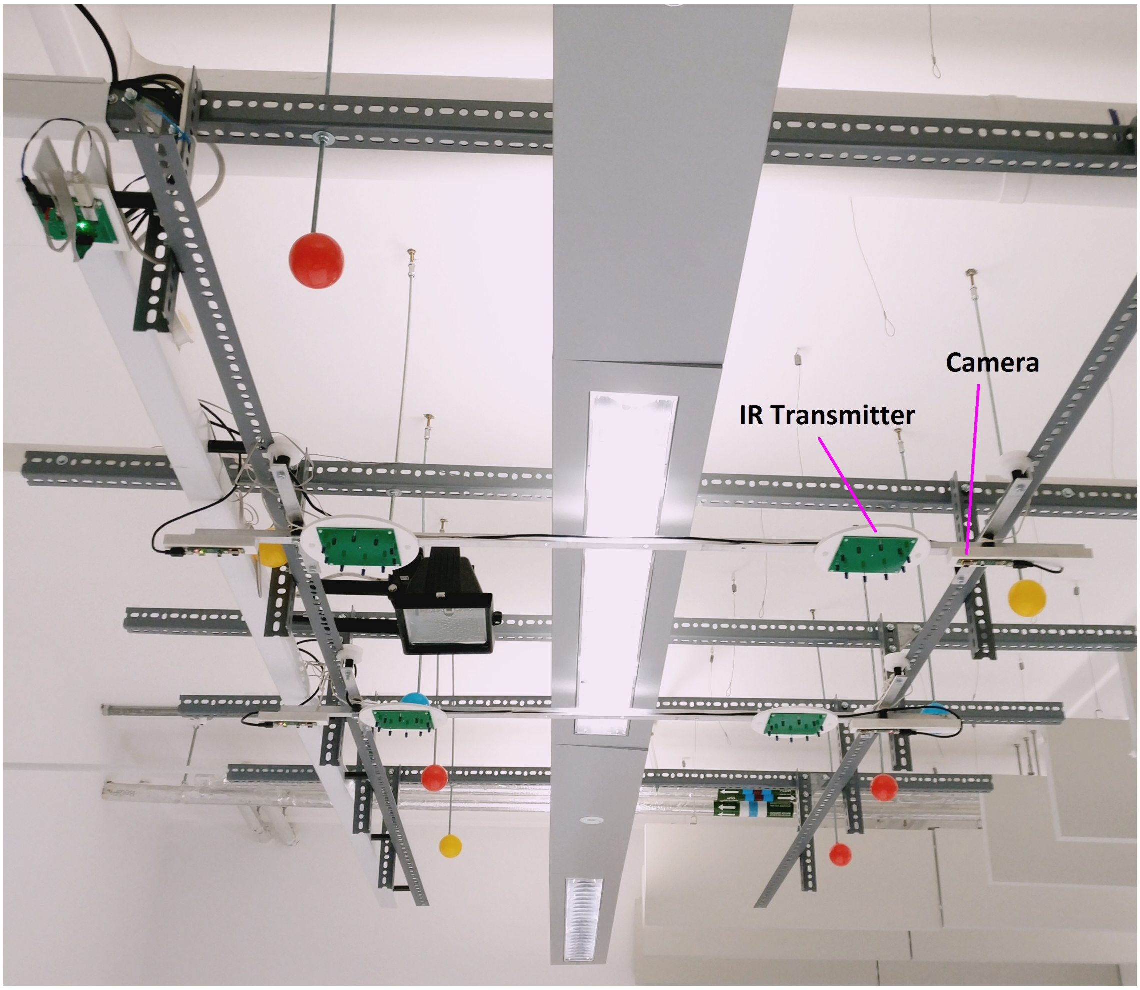


Figure 4.2: ARK technology: there are four cameras used to record the experiment in the underlying arena. Several IR transmitter PCBs are used to send messages to the Kilobots during the experiment.

4.2 Straight motion by robots of different size

This experiment is an analysis of the performance of Kilobot Soft Robots of different sizes when instructed to keep a straight motion. The aim is to evaluate any kind of relationship that exists between the Soft Robot size S and the coordinated motion in terms of trajectory stability and speed. The Robot is tasked to move straight forward for 120 cm without any feedback by the ARK system about its position or trajectory. This means the Kilobot Soft Robot is controlled in an open-loop setting. However, among the individual modules a feedback control in implemented to keep the system shape: the Kilobots estimate the neighbours relative positions and act accordingly. In this experiment we test different Soft Robot sizes S, where $S \in \{1\times1, 2\times2, \ldots, 7\times7\}$, while measuring the deviation from the ideal trajectory and the mean travel speed as explained in the *experimental results* subsection.

4.2.1 Experimental setup

We conducted experiments with robots of 7 sizes, $S \in \{1 \times 1, 2 \times 2, \ldots, 7 \times 7\}$. For each size, 10 trials were performed, that is, 70 trials in total. At the beginning of every run the Kilobot Soft Robot orientation and position are set taking as reference point its centre of mass (CoM, indicated by a red star in Figure 4.3), that is the average position of all Kilobots composing the Robot. Each trial was run for a fixed duration, T. If the robot (its CoM) reached the finish line, the trial is considered successful, and stopped. Otherwise, the trial is considered unsuccessful and stopped after $T = 800\,\mathrm{s}$, except for $S = 1 \times 1$ and $S = 2 \times 2$, where runs are stopped after a maximum duration of $T = 400\,\mathrm{s}$ and $T = 600\,\mathrm{s}$, respectively. The duration T is chosen as the double of the time required to complete successfully a trial (in the absence of any faults) which is established through preliminary tests.

The Figure 4.3 is a top view of the experimental arena showing the result of a trial with a 7×7 Kilobot Soft Robot. The first and last frame of the run are superimposed together with the experimental setup comparing the initial and final configuration in terms of lateral deviation and shape distortion. In the starting position the Soft Robot is oriented in parallel to the x-axis to directly face the finish line (on the right) and its CoM positioned on the centre of the vertical starting line (left dotted line in Figure 4.3). The starting and finishing lines (the dotted ones) are positioned at 40 cm from the arena boundary at a distance of 120 cm from each other.

During the simulation all the Kilobots IDs and coordinates, taken relative to a reference system fixed on a corner of the arena, are recorded over time. These data allow the analysis, discussed in the following Section, of speed, shape distortion and lateral deviation as functions of the Kilobot Soft Robot size S.

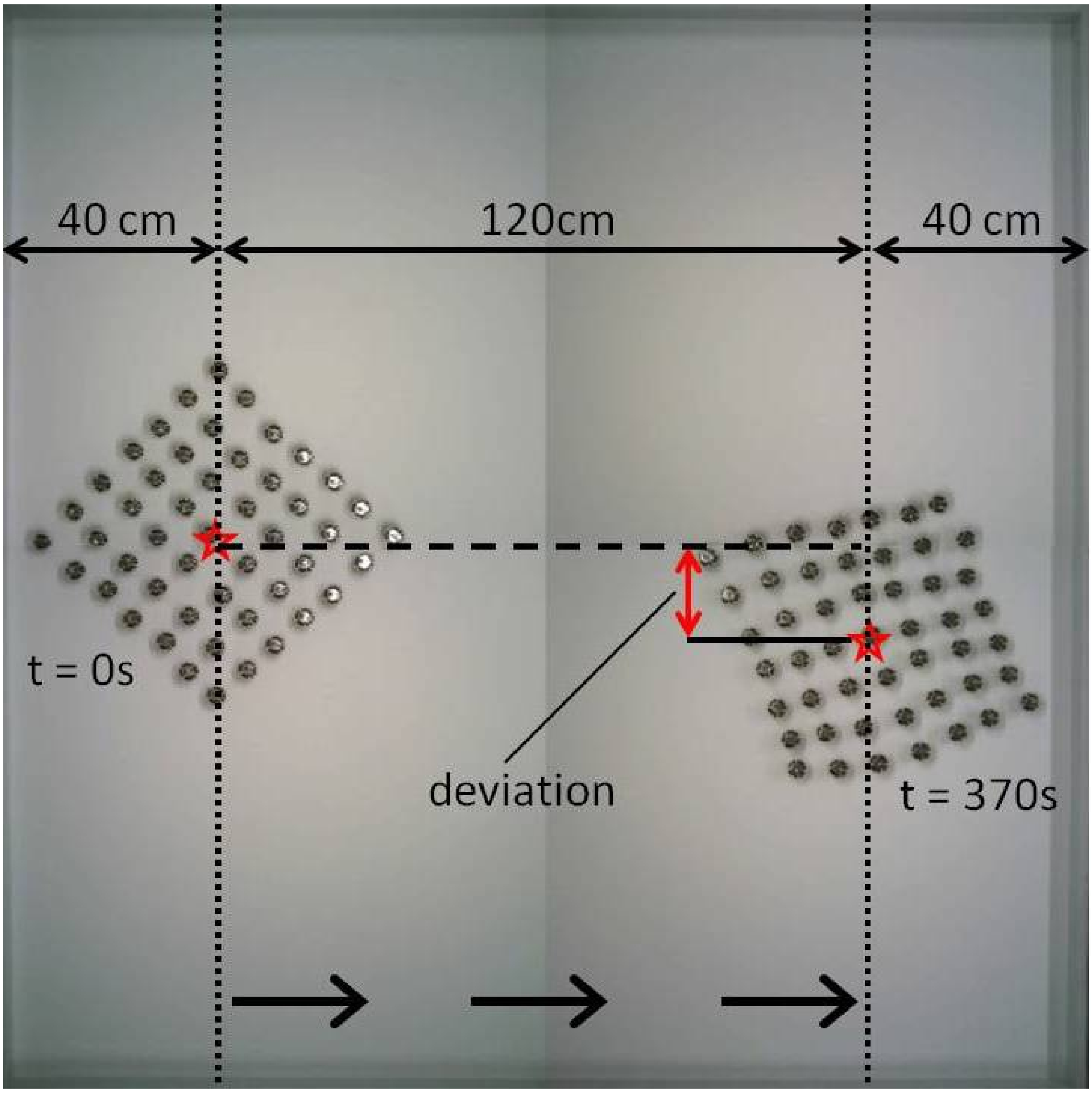


Figure 4.3: A top view of the experimental arena showing the result of a trial with a 7×7 Kilobot Soft Robot. The starting and final frame are superimposed together with the experimental setup to illustrate the method followed for the experiments. The vertical dashed lines represent the starting and finishing points. The red star indicated the Robot Centre of Mass, taken as reference point during the experiments. Once the robot reaches the finishing line, its lateral deviation and shape distortion are recorded.

4.2.2 Experimental results

All the experimental data have been analysed and plotted using *matplotlib*, python library. However, before any graph, a clear result can be observed from the percentage of successful trials in the experiments for each Kilobot Soft Robot size, S:

larger is the Robot, the higher is the probability that it reaches the finish line (see Table 4.1).

Table 4.1: Percentage of successful trials in the experiments where a Kilobot Soft Robot, of size S, is tasked to move forward by 120 cm in the absence of any external feedback. 10 trials per robot size.

Robot size, S	Success rate (in percent)
1×1	20
2×2	30
3×3	80
4×4	90
5×5	90
6×6	90
7×7	90

The same conclusion can be reached looking at the Figures 4.4 and 4.5, which show the trajectories taken by the smallest and largest robots, respectively, in all trials. In both cases the Soft Robot is tasked to keep a straight motion direction without an external feedback from the ARK system. The difference between the two Robot configurations is evident: the 1×1 one, essentially a single Kilobot, is unable to keep a constant direction differently from the 7×7 one which is able to deviate far less from the desired trajectory. The increasing error on the motion, which the single Kilobot suffers, can be compensated when there are more modules that cooperate and work as feedback for their neighbours. The error in the direction followed is reduced and the mobility accuracy is improved but the Soft Robot will not stay on course indefinitely due to the lack of an external feedback (open-loop control).

Hence, there will be always a deviation from the desired trajectory whose entity depends on the Robot size S. This feature is shown in Figure 4.6. Here, we focus on the successful trials. Once the robot (CoM), tasked to move forward by a distance of 120 cm, reaches the finishing line, the trial is successfully completed. At this time we extract from the data the lateral deviation of the Robot centre of mass from the ideal target point, as shown in Figure 4.3. The scatter diagram, shown in Figure 4.6, illustrates the absolute deviation of successful runs as a function of the Kilobot Soft

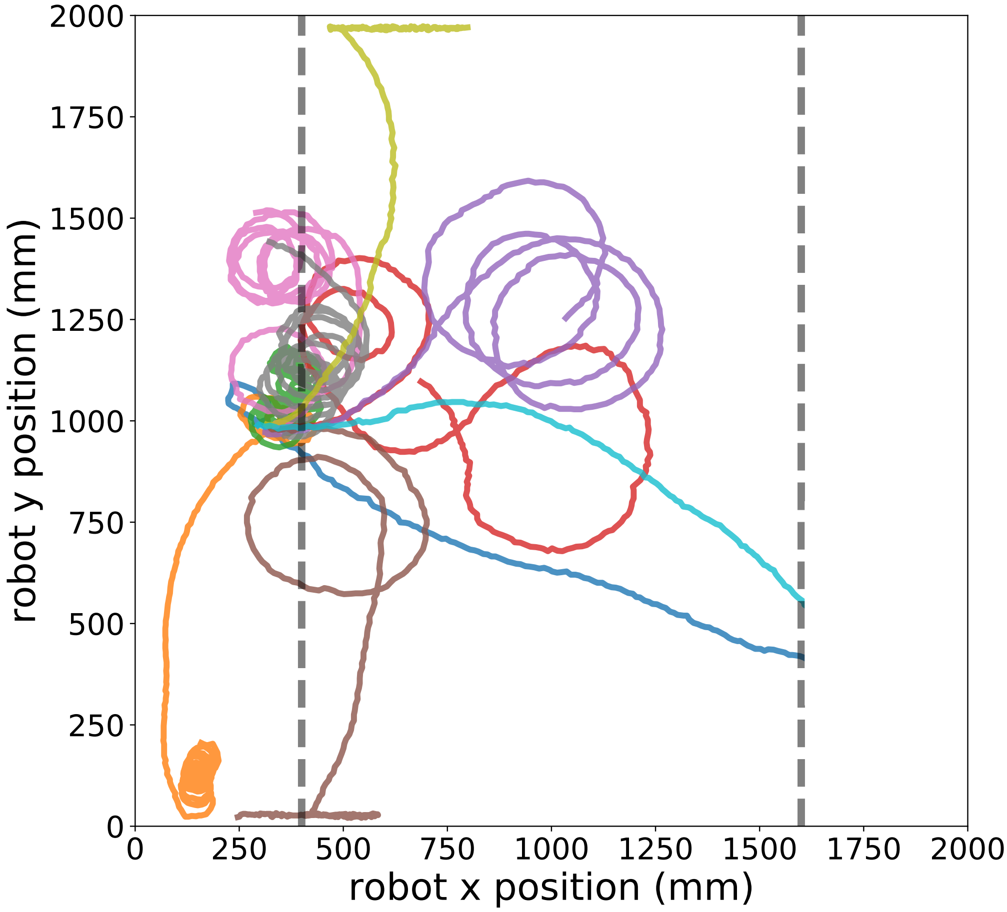


Figure 4.4: Trajectories of 1×1 Kilobot Soft Robots. Substantially, a single Kilobot is tasked to move straight without an external feedback. The dashed lines represent the starting and finishing positions. In two trials, out of ten, the Robot completed the task, reaching the finishing line.

Robot's size S. For each size, we report on the top the number of failed trials in which the Robot did not reach the finishing line within the maximum run time.

The mean value (the blue line graph) shows a decreasing trend in deviation for increasing sizes of S. In other words, the motion of the Kilobot Soft Robot becomes more accurate, the more modules it has. The sensing and actuation of individual Kilobots are highly noisy and the two successful trials show a high irregularity in the path tracked (Figure 4.4). When a large number of Kilobots are gathered together these inaccuracies seem to be compensated by a collective cooperation. The paths tracked by the Robots are more coherent, the gap between the lateral deviation of different runs becomes smaller.

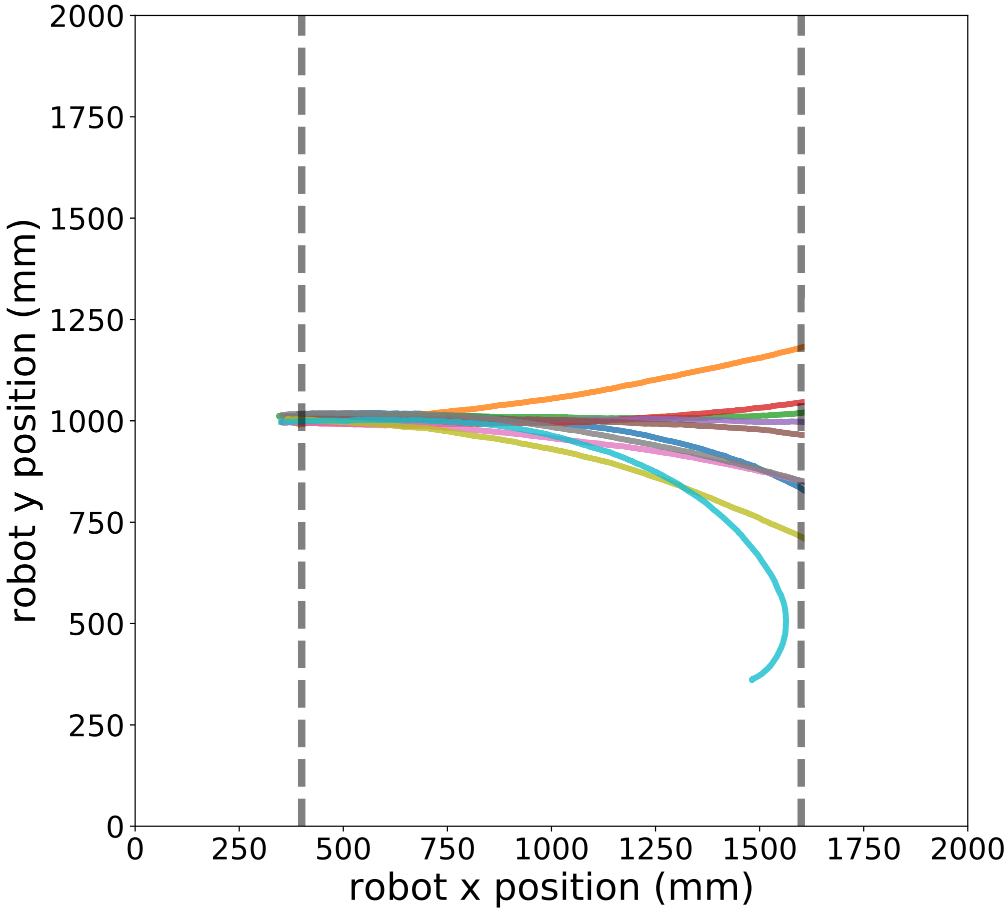


Figure 4.5: Trajectories of 7×7 Kilobot Soft Robots when tasked to move straight without an external feedback. Dashed lines represent the starting and finishing positions. In one trial, out of ten, the Robot failed to complete the task, nearly reaching the finishing line.

Another feature analysed as function of the Kilobot Soft Robot size S is the speed, as shown in Figure 4.7. The scatter diagram shows how a performance improvement in terms of accuracy corresponds a reduction of the speed kept on average by the Soft Robot. Processing the data files the speed was computed as the ration between the sum of the Euclidean distances of the CoM's position every $\Delta t = 1\,\mathrm{s}$ and the total experiment length, among the successful runs only. Interestingly, while accuracy tends to improve with robot size S, the speed seems to settle to a constant value of about $0.35\,{}^{\mathrm{cm}}\!/\!_{\mathrm{s}}$. Considering the individual Kilobots' speed of $\sim 1\,{}^{\mathrm{cm}}\!/\!_{\mathrm{s}}$, we can say that the Kilobot Soft Robot, when constituted by a large number of modules, has a speed of about 35% the one of its individual constituent robots.

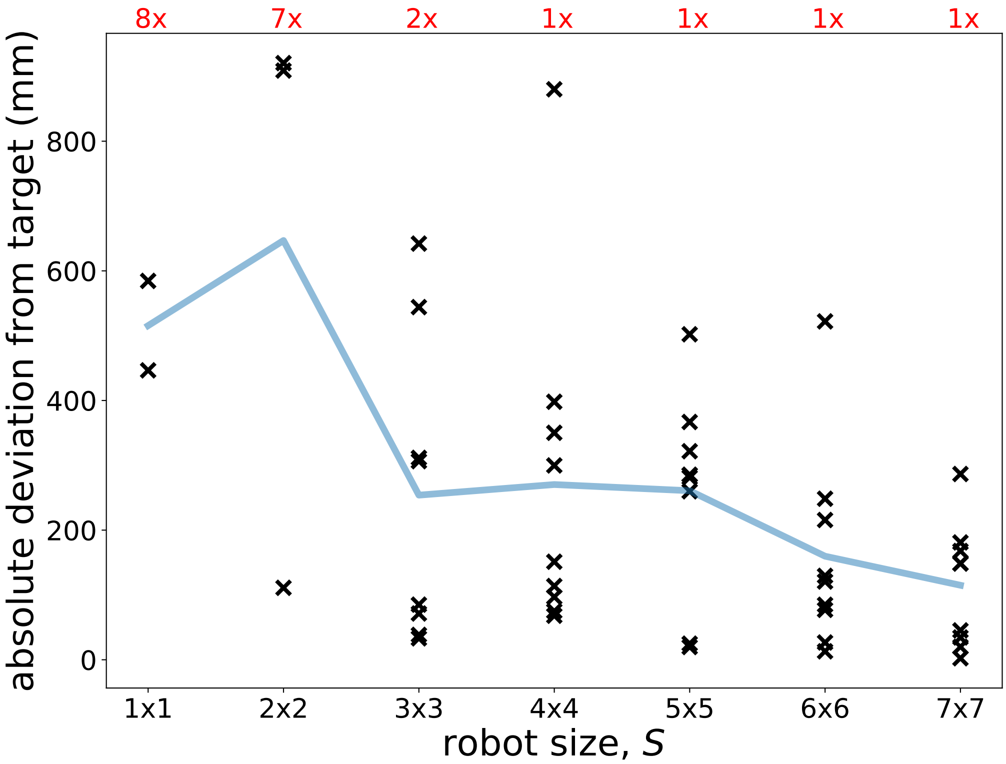


Figure 4.6: Motion accuracy of Kilobot Soft Robots of size $S \in \{1{\times}1, 2{\times}2, \ldots, 7{\times}7\}$ when tasked to move straight for 120 cm without an external feedback. Each mark " $\times$ " indicates the absolute lateral deviation from the target (see Figure 4.3) of a successful run. The number of unsuccessful runs are reported on top in red. The blue line represent the mean value. Increasing the size S the probability the Robot reaches the finishing line increases.

Finally, for each trial, we computed the root-mean-square distortion as a coefficient to quantify the robot distortion from the target shape, the regular square lattice one as shown in Figure 3.1. A scatter diagram is generated to plot the distortion coefficient as a function of the Kilobot Soft Robot size S for both successful and unsuccessful trials (see Figure 4.8). This is a coefficient that varies over time and is computed as the mean squared error from 90 degrees for all the internal angles. Indeed, these angles are all the ones formed between two adjacent links for each module and are, in the square lattice target shape, all at 90 degrees. For example, in a 3×3 Robot the mean squared error is computed over time for sixteen angles.

The distortion values plotted in the scatter diagram are the ones at the time the simulation is stopped. Moreover, in Figure 4.8 the average of the distortion

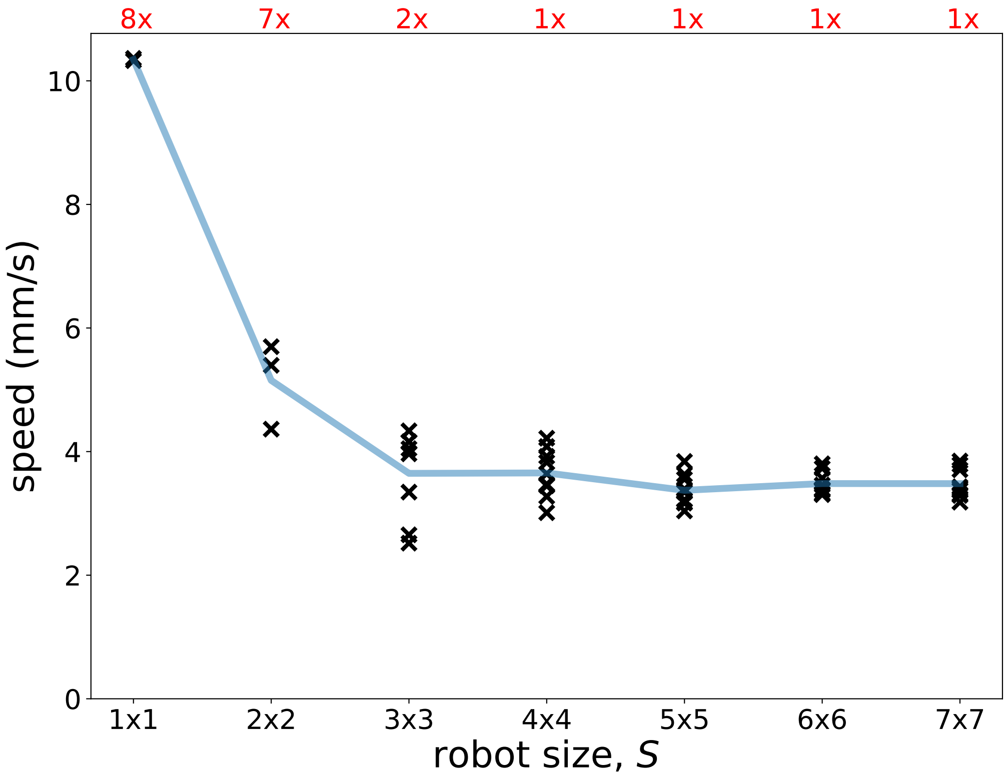


Figure 4.7: Speed of Kilobot Soft Robots as a function of their size $S \in \{1 \times 1, 2 \times 2, \ldots, 7 \times 7\}$ when tasked to move straight for 120 cm without an external feedback. Only the successful runs are considered and the number of unsuccessful ones are indicated on top in red. The blue line indicates the mean value. Small groups are quicker but probability to reach the finish line is lower.

coefficient for each Robot size is reported over time (blue line).

The distortion coefficient for any size in almost all the successful trials (black circles) is relatively low, while we observe a large distortion in some of unsuccessful trials (red crosses). In these cases the failure of the run is caused by a high proportion of elastic links and modules that move in a wrong configuration ruining the robot mobility. In the cases of unsuccessful trials with a low distortion coefficient the Kilobot Soft Robot could not cross the finish line due to the too big lateral deviation.

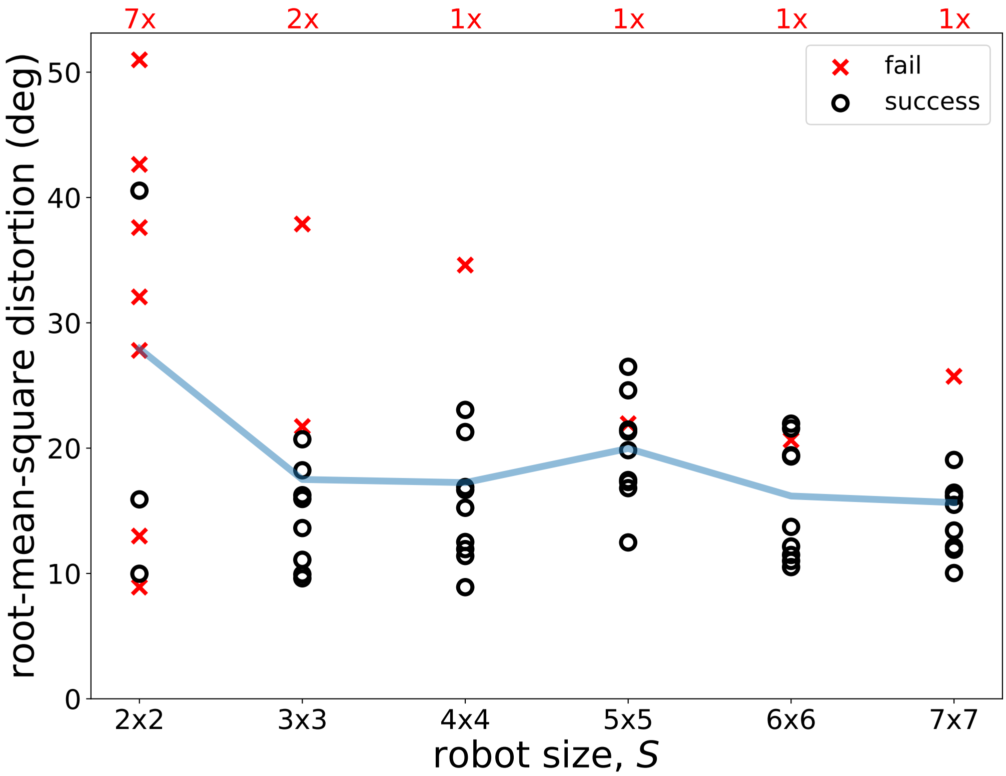


Figure 4.8: Kilobot Soft Robot shape distortion from the target lattice one as a function of the size $S \in \{2\times 2, \dots, 7\times 7\}$. The black circles and red crosses represent respectively both successful and unsuccessful trials. The blue line indicates the mean value of all the distortion values for each trial.

4.3 Trajectory following

This experiment quantifies the performance of Kilobot Soft Robots when instructed to follow a predefined, curved trajectory. The frontal module (at position P_1 in Figure 3.8) is provided with feedback from the ARK system according to the Soft Robot centre of mass position. At 2 s intervals, it receives one bit of information, indicating whether the reference trajectory is to the left or to the right of the Robot (its CoM) in order to correct its motion direction.

4.3.1 Experimental setup

The reference trajectory is a circle of radius 70 cm virtually placed in the centre of the arena as illustrated in Figure 4.9. The ARK system, tracking the Kilobots in the arena, checks the Soft Robot CoM position relative to the trajectory virtually drawn in the ARK system itself. A boolean value is transmitted to modules, indicating whether the CoM is inside or outside the circle, respectively to the right or to the left of the reference path (counterclockwise direction). Only the front Kilobot uses this information to correct the trajectory followed by the Soft Robot, since its motion can influence the one of the whole system.

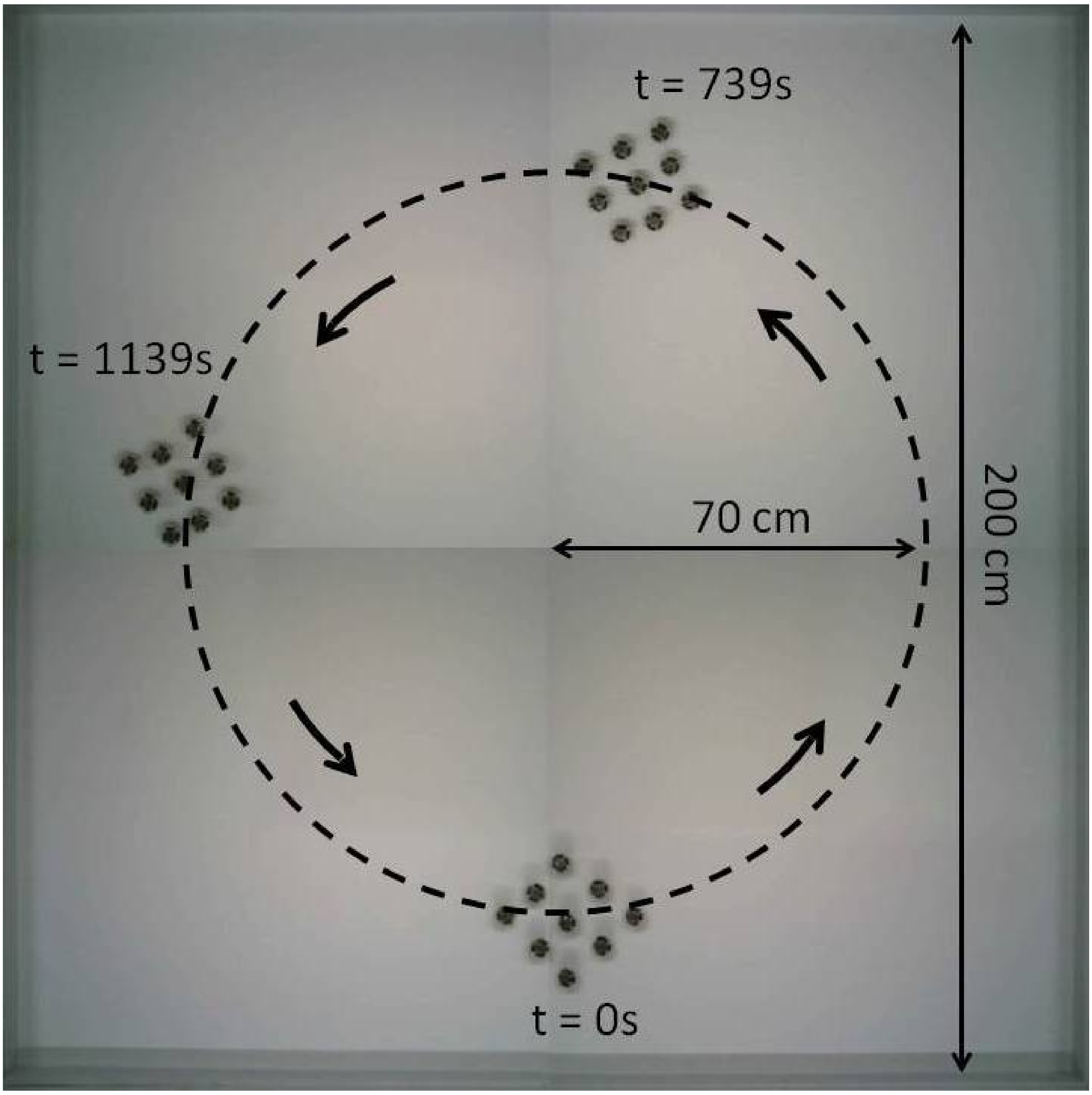


Figure 4.9: A top view of the experimental arena showing the result of a trial with a 3×3 Kilobot Soft Robot, where the task is to complete a circular path (dashed circle) of radius 70 cm. Three frames at different simulation times are superimposed together with the experimental setup and the circular trajectory followed.

At the beginning of every trial the 3×3 Kilobot Soft Robot is initially placed on the circular path, in a counterclockwise direction. In this experiment five trials were performed. Each one has a maximum run-time duration, chosen as the double of the expected time required to complete successfully (without any faults) the run and established through preliminary tests. If the robot (its CoM) completes one full revolution on the desired trajectory, the trial is considered successful, and stopped. Otherwise, the trial is considered as unsuccessful. Moreover, runs where the robot collides with the arena boundary are aborted, and considered unsuccessful.

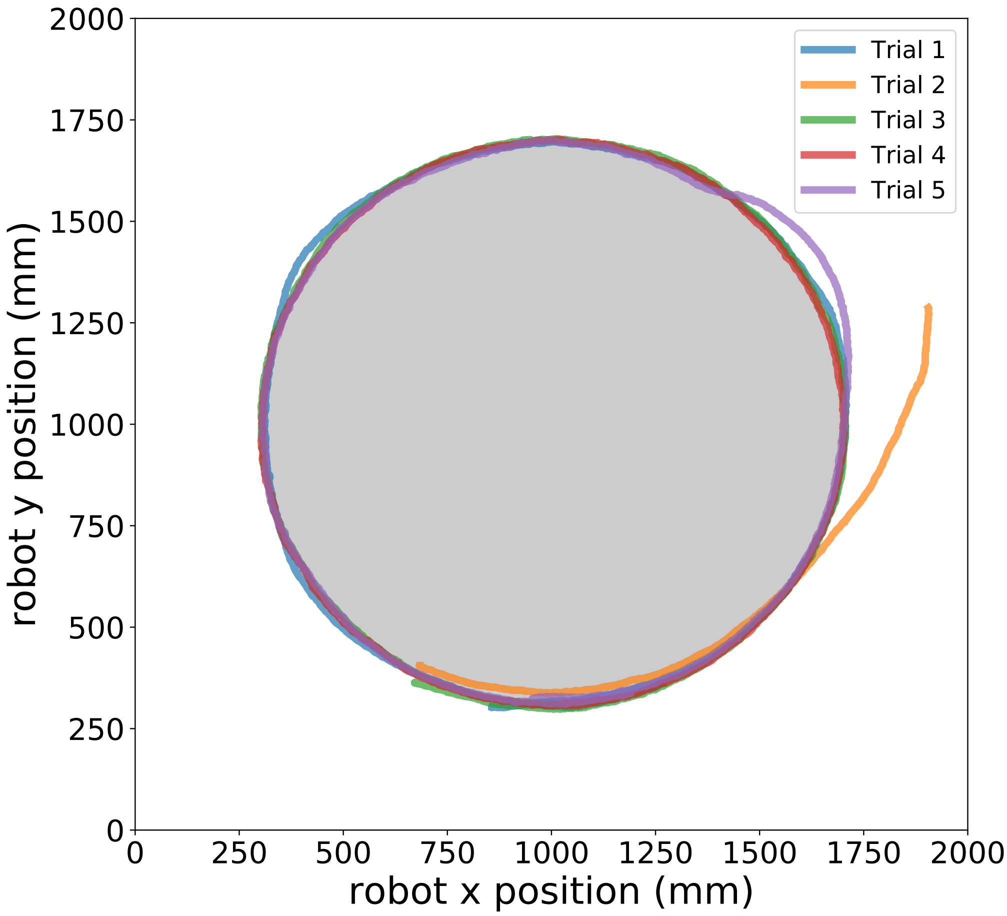


Figure 4.10: Motion trajectories of the Centre of Mass (CoM) of the 3×3 Kilobot Soft Robot when tasked to follow a circular trajectory with closed-loop control. The ideal path is represented by the dark circle, the coloured lines represent the five trials of the experiment. Four times, out of five, the Soft Robot successfully completed the revolution following quite accurately the trajectory, one time (trial 2) the Robot failed the task as hitting the right of the arena boundary.

4.3.2 Experimental results

Looking at Figure 4.10, which shows the trajectories for each trial, is possible to see the ability of the system to follow the circular trajectory. One trial (Trial 2, yellow line), out of five, was not successful, as the robot hit the arena boundary and could not complete a round. In all other trials the Soft Robot CoM follows the reference trajectory with reasonable accuracy.

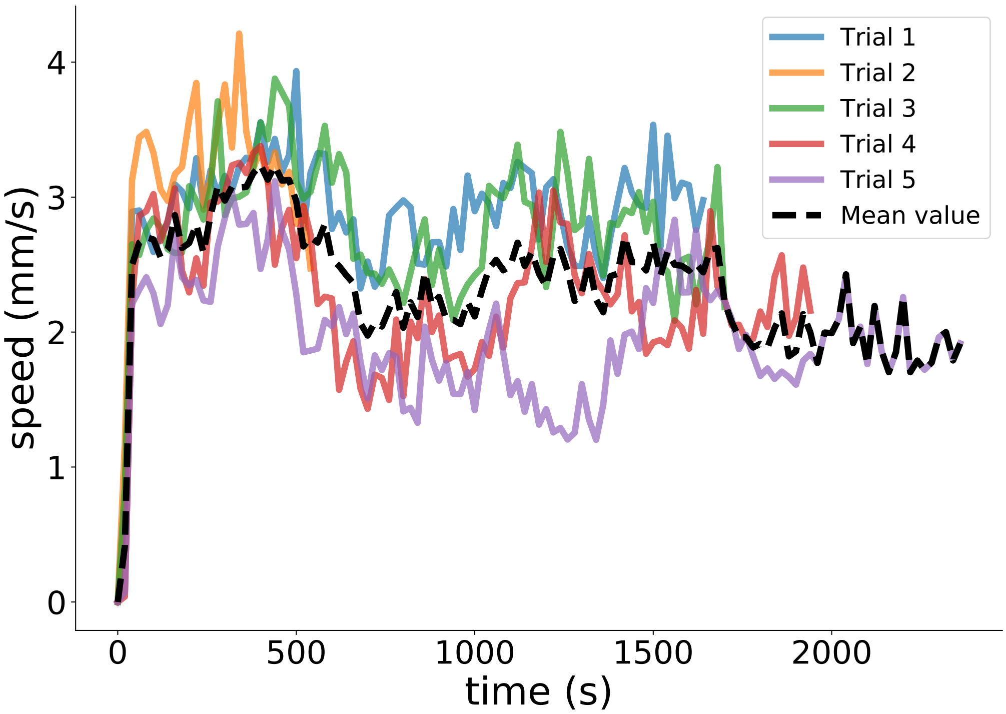


Figure 4.11: Linear speed of the 3×3 Kilobot Soft Robot (its CoM) when tasked to follow a circular trajectory with closed-loop control. The dashed line represent the average value of the trials speed over time (coloured lines). The yellow line (Trial 2) stops long before the others because it refers to a trial that failed, as the Soft Robot hit the right of the arena boundary.

Another feature analysed is the linear speed of the Soft Robot (its CoM) over the duration of the five trials (see Figure 4.11). The coloured lines represent the velocities over time for each trial, while the black line represents the mean value. The speed values were computed with a finite resolution (10 s intervals) to reduce the oscillations in the graph. The results show that the Soft Robot, following the desired trajectory, moves roughly with a constant speed. The yellow line stops around 500 s, as it represents the unsuccessful trial.

4.4 Deforming the shape while following a trajectory

This experiment quantifies the performance of the Kilobot Soft Robot when instructed to deform the shape of its body while following a predefined trajectory. In particular, the ability to squeeze and extend the shape at a certain moment is tested. The robot first moves keeping its default shape, then shrinks and finally expands again, to restore the original shape. As previously, a feedback is used through the ARK system to control the direction followed by the swarm, to simulate a hypothetical situation, in which the Soft Robot is able to reach a final goal, thanks to the ability of deforming its shape to avoid some obstacles. Two signals are sent by the ARK system. One is an instruction for all the modules to indicate when the shape change has to be triggered. The other, periodically sent (at 2 s intervals) for the frontal module (at position P_1 in Figure 3.8), is a boolean value, indicating whether the reference trajectory is to the left or to the right of the robot centre of mass, considering the system motion direction.

4.4.1 Experimental setup

We used a 4×4 Kilobot Soft Robot, and conducted 11 trials. Similar to the experiment discussed in Section 4.3, a feedback is provided, via IR communication, to the Robot to inform the frontal module about the current position of the CoM, with respect to the desired trajectory. This is a straight line from the initial to the final position, aligned with the x-axis, shown as horizontal dashed line in the Figure 4.12. This is a top view of the experimental arena in which three frames, taken at different times, one from each deforming phase, are superimposed together with the experimental setup. The initial and final positions for the robot (CoM) are set respectively at 40 cm and 180 cm from the left of the arena.

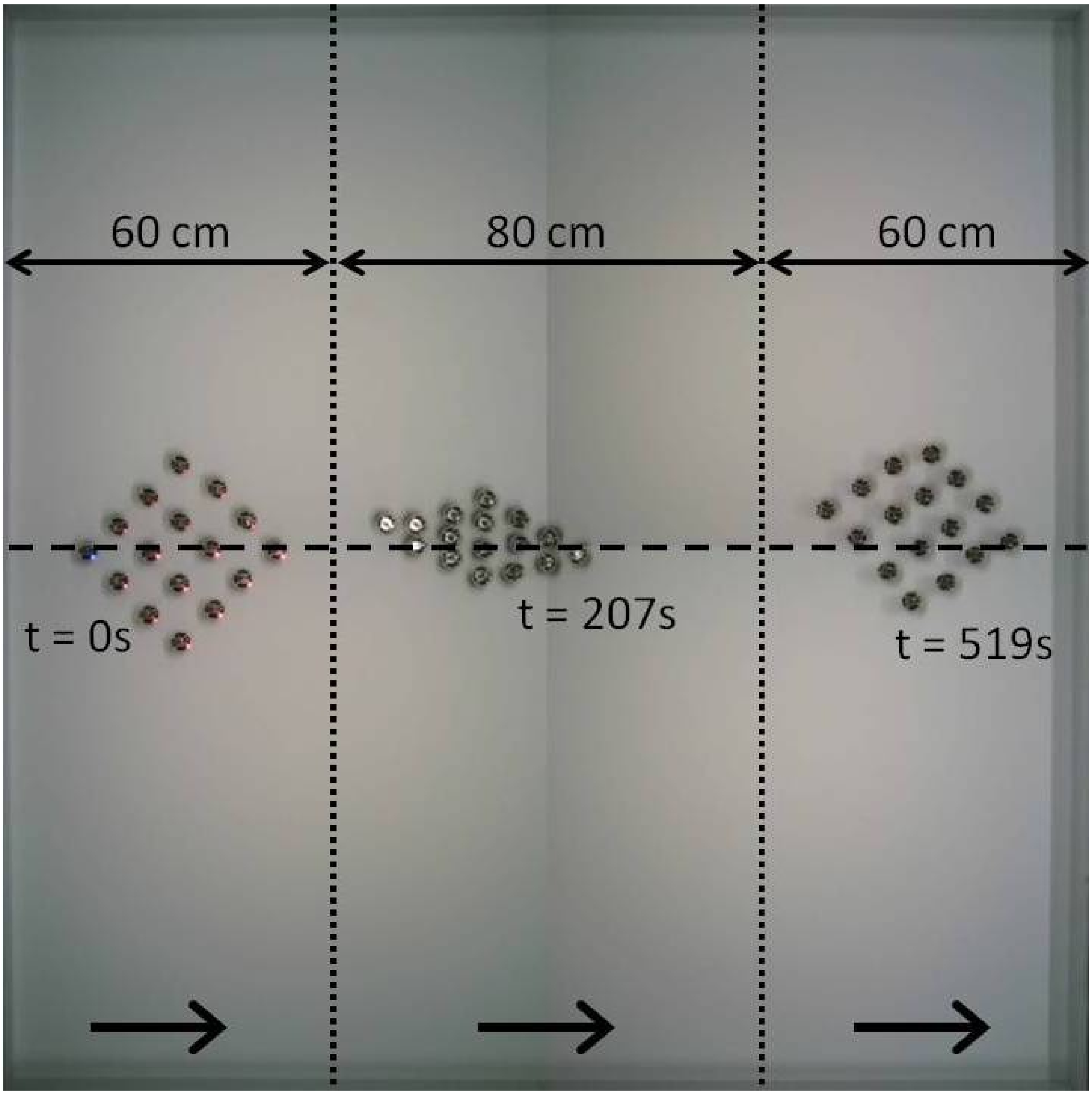


Figure 4.12: A top view of the experimental arena showing the result of a trial where a 4×4 Kilobot Soft Robot is tasked to advance forwards while, starting from an extended shape, first compressing its shape and then restoring it to its original form. Three frames, one from each deforming phase, are superimposed together with the experimental setup. The vertical dotted lines represent the positions where shrinking (left line) and expanding (right line) procedures start.

The ARK system is also exploited for sending messages that trigger the shape deformation. When the vertical dotted lines (see Figure 4.12), positioned at 60 cm from the left and 60 cm from the right, are crossed by the frontal module P_1 (see Figure 3.8) the commands to first shrink and then extend the body are sent respectively to all the modules of the Soft Robot.

Each trial is run for a fixed duration, $T = 1000\,\mathrm{s}$. If the robot centre of mass reached a distance smaller than 20 cm from the right boundary, the trial was considered successful and stopped. Otherwise, the trial was considered unsuccessful. The duration T is chosen, as previously, at least the double of the time required to complete successfully (in the absence of faults) a trial. This time is established through preliminary test.

4.4.2 Experimental results

Figure 4.13 shows, for each trial, the distortion with respect to the regular lattice shape, measured during the experiment. As in the previous Section 4.2, the distortion parameter is computed as the mean squared error from 90 degrees for all the angles among the modules. High values of this parameters indicate that several angles are significantly different from 90 degrees, which means the body of the Kilobot Soft Robot is shrunk. Low values of this parameter, instead, indicate that the system is keeping the lattice square shape (the target shape). The following graph illustrates the variation of the distortion parameter as function of the frontal module movement.

The Soft Robot at the starting point is in a regular lattice shape, with low values of the distortion parameters. It starts moving until the left dotted line (at 60 cm) is crossed by the frontal Kilobot, the instruction that triggers the deformation of the shape is sent by the ARK system and the distortion coefficient rapidly increases (see Figure 4.13) almost to the reference value of $\alpha = 50$ degrees (see the *deformation* algorithm Section). Subsequently, the Soft Robot moves keeping the shrunk shape, following a straight direction thanks to the closed loop control through ARK system, until the right dotted line (at a distance of 140 cm) is crossed, the instruction to restore the Robot shape is sent and the distortion coefficient rapidly decreases.

The shape change is relatively quick and is completed within a travelled distance of $10 \sim 20$cm.

Moreover, it is worth noting that, of 11 trials, only in one trial the Soft Robot did

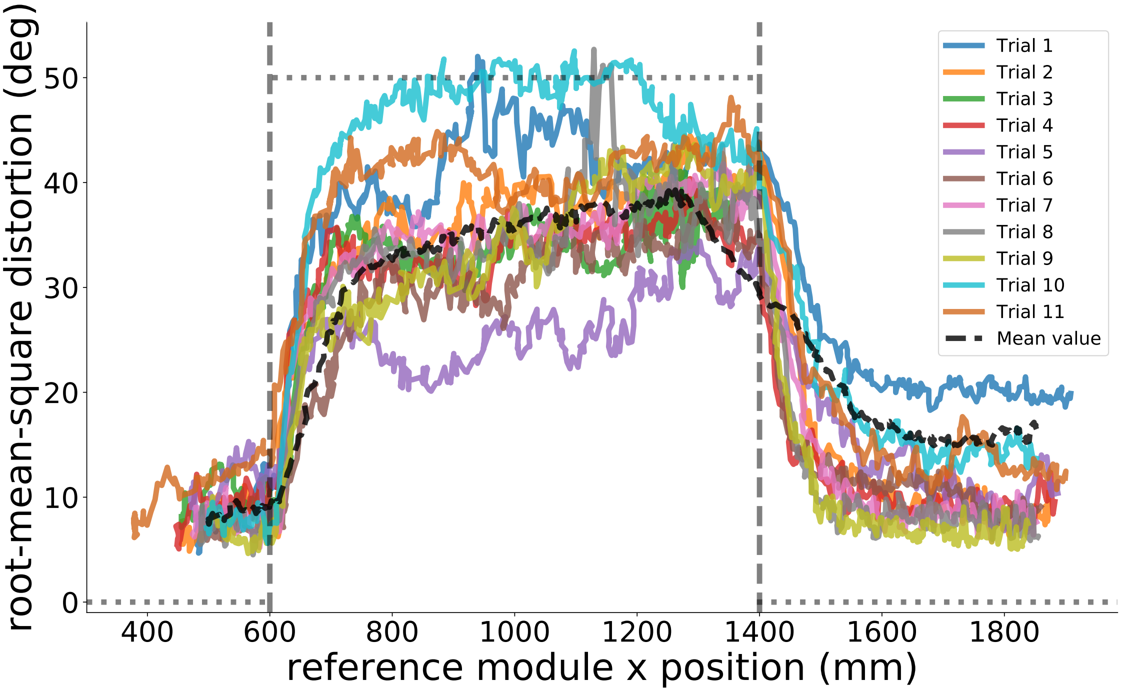


Figure 4.13: The shape distortion of a 4×4 Kilobot Soft Robot that is tasked to move forward while deforming its shape. The target shape is a square lattice for the first 60 cm, a stretched lattice from 60 cm to 140 cm, and the original shape thereafter. During the first phase the *motion control* algorithm is implemented, with the offset parameter $\alpha = 0$. During the second phase the *deformation* algorithm implemented, with the offset parameter $\alpha = 50$. During the last phase the shape is restored through the *motion control* algorithm. Shown are the root-mean-square distortion values observed in the 11 trials. All trials are successful except Trial 3, which could not reach the second dotted line. The black dashed line represents the mean value, while the grey dotted line the reference value (α).

not reach the finish line (Trial 3 in Figure 4.13). In this trial, the frontal module, due to a poor calibration of one motor or a low battery level, failed to pull the Robot properly to follow the reference trajectory until the target destination. The Robot could not reach the second dotted line and remained in the central part of the experimental arena in a squeezed shape (green line in the graph). This type of errors could be avoided through the use of more than one leading module, as discussed in the following Section. Indeed, the control on the trajectory was implemented, in this and previous experiments, through a communication between the ARK system and the frontal module, which played the role of ”leader” in the group of Kilobots.

Chapter 5

Discussion and conclusion

In this dissertation we presented the Kilobot Soft Robot, a novel concept for a modular soft-bodied robot. The Kilobot Soft Robot is composed of simple mobile modules, Kilobots, which are interconnected by springs to create a deformable 2-D lattice structure. The modules can be manually reconfigured to create lattices of arbitrary dimensions.

We presented a control algorithm that allows the Kilobot Soft Robot to move in a planar environment, while controlling its shape. The proposed algorithm is fully distributed: each module has a battery, sensors, computational capabilities, and an actuation system, and makes its decisions based on locally available information.

The Soft Robot is characterised by different capabilities. In particular, the control algorithms developed allow the system to move forward while keeping a constant shape, to change the shape (compressing and decompressing) during the motion, and to follow a reference trajectory.

A series of experiments was conducted to analyse the performance of the realized Robot, increasing its size up to 49 modules, organised in a 7×7 lattice. Several trials were performed for each experimental scenario.

The results show that the accuracy of the robot's motion increases considerably with the number of modules. While a single module, or small-sized Soft Robots, are inaccurate, influenced by noise and motion errors, Robots with a large number of modules allow to compensate the errors introduced by the Kilobots mobility making the motion more accurate.

However, at the same time, the increasing accuracy is paid with the speed that decreases with the Robot size. However, the experiments results show that the speed settles roughly to a constant value for sizes larger than 4×4. We can conclude, hence, that it is possible to increase the accuracy without excessively compromising the speed.

As a conclusion, the results discussed in this dissertation represent the one of the first implementations of a soft-bodied robot composed by a large number of fully autonomous modules, that can be arbitrarily reconfigured. Moreover, this study presents also one of the first rigorous experimental evaluations of a self-propelling modular robot at a large scale (49 modules). Hence, in view of the results obtained and here discussed, the question if soft connections are superior to rigid ones, for modular robots that move across the ground, is born. This is an issue that could be tested in the future.

One of the main advantages of this system is the modular and distributed approach, through which a good fault tolerance is automatically implemented. Indeed, computation and motor actuation are totally decentralised, hence the system is expected to compensate failures with possibly only slightly decreased performance. However, in the closed-loop experiments, our current implementation of the motion algorithm relies on a leader module (the frontal one), which brings the Soft Robot to follow the reference trajectory. This module is in charge of influencing the overall motion of the Kilobot Soft Robot through the effects of the springs and the detection of the distances by its neighbours. A single module is sufficient to guide the movement of a relatively large group towards a desired direction, but, at the same time, it represents a cause of failure, as happened in the deformation experiment, where an error on the frontal module prevented the system from completing its task.

In the future, we aim to improve the system, enabling all modules to receive and integrate the external feedback. Future work will aim at reformulating the algorithm, in order to avoid the presence of a single point of failure, taking a cue from the adaptive behaviours observed in biological systems, such as the frequent leader switching in groups of ants.

Another notable future work may be to consider the presence of external objects and an interaction between them and the Kilobot Soft Robot. In particular, it could be possible to improve the Robot with the ability to detect the presence, and even understand the shape of these objects, analysing the internal deformations among the modules. The Kilobot platform is not provided of any contact sensor, but this feature can be integrated to the Soft Robot thanks to a cooperation of all the modules without adding any external sensor. Potentially these objects could be also moved and manipulated.

The studies, activities and the results investigated in this dissertation have been used to write a scientific paper for a journal in collaboration with researchers and swarm robotics experts. The article, recently submitted to the IJRR (International Journal of Robotics Research) special issue "Soft Robotic Modeling and Control: Bringing Together Articulated Soft Robots and Soft-Bodied Robots" and now under review, has the title: Federico Pratissoli, Andreagiovanni Reina, Yuri Kaszubowski Lopes, Lorenzo Sabattini, Roderich Groß (2019) The Kilobot Soft Robot: A Soft-Bodied Modular Reconfigurable Robotic System.

Bibliography

Albu-Schäffer A and Bicchi A (2016) Actuators for soft robotics. In: *Springer Handbook of Robotics*. Springer, pp. 499–530.

Baldassarre G, Trianni V, Bonani M, Mondada F, Dorigo M and Nolfi S (2007) Self-organized coordinated motion in groups of physically connected robots. *IEEE Transactions on Systems, Man, and Cybernetics, Part B* 37(1): 224–239.

Brochu P and Pei Q (2010) Advances in dielectric elastomers for actuators and artificial muscles. *Macromolecular rapid communications* 31(1): 10–36.

Gazi V and Passino KM (2004) A class of attractions/repulsion functions for stable swarm aggregations. *International Journal of Control* 77(18): 1567–1579.

Gilpin K and Rus D (2010) Modular robot systems. *IEEE Robotics & Automation Magazine* 17(3): 38–55.

Hamlin GJ and Sanderson AC (1995) Tetrobot: a modular system for hyper-redundant parallel robotics. In: *IEEE International Conference on Robotics and Automation (ICRA)*, volume 1. IEEE, pp. 154–159.

Hirose S, Shirasu T and Fukushima EF (1996) Proposal for cooperative robot "Gunryu" composed of autonomous segments. *Robotics and Autonomous Systems* 17(1-2): 107–118.

Kwok SW, Morin SA, Mosadegh B, So JH, Shepherd RF, Martinez RV, Smith B, Simeone FC, Stokes AA and Whitesides GM (2014) Magnetic assembly of soft robots with hard components. *Advanced Functional Materials* 24(15): 2180–2187.

Lee JY, Kim WB, Choi WY and Cho KJ (2016) Soft robotic blocks: introducing sobl, a fast-build modularized design block. *IEEE Robotics & Automation Magazine* 23(3): 30–41.

Leonard NE and Fiorelli E (2001) Virtual leaders, artificial potentials and coordinated control of groups. In: *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, volume 3. IEEE, pp. 2968–2973.

Mathews N, Christensen AL, O'Grady R, Mondada F and Dorigo M (2017) Mergeable nervous systems for robots. *Nature communications* 8(1): 439.

Murray L, Timmis J and Tyrrell A (2013) Modular self-assembling and self-reconfiguring e-pucks. *Swarm Intelligence* 7(2-3): 83–113.

Onal CD and Rus D (2012) A modular approach to soft robots. In: *IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, pp. 1038–1045.

Reif JH and Wang H (1999) Social potential fields: A distributed behavioral control for autonomous robots. *Robotics and Autonomous Systems* 27(3): 171–194.

Reina A, Cope AJ, Nikolaidis E, Marshall JAR and Sabo C (2017) ARK: Augmented Reality for Kilobots. *IEEE Robotics and Automation Letters* 2(3): 1755–1761.

Reynolds CW (1987) Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH Computer Graphics* 21(4): 25–34.

Rubenstein M, Ahler C, Hoff N, Cabrera A and Nagpal R (2014) Kilobot: A low cost robot with scalable operations designed for collective behaviors. *Robotics and Autonomous Systems* 62(7): 966–75.

Rubenstein M, Ahler C and Nagpal R (2012) Kilobot: A low cost scalable robot system for collective behaviors. In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 3293–3298.

Salisbury K, Townsend W, Ebrman B and DiPietro D (1988) Preliminary design of a whole-arm manipulation system (wams). In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 254–260.

Talamali MS, Bose T, Haire M, Xu X, Marshall JAR and Reina A (2019) Sophisticated collective foraging with minimalist agents: A swarm robotics test. *under review* .

Wang W and Ahn SH (2018) Mechanical assembly of soft deployable structures and robots. In: *IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, pp. 222–227.

Yim M, Shen WM, Salemi B, Rus D, Moll M, Lipson H, Klavins E and Chirikjian GS (2007) Modular self-reconfigurable robot systems [grand challenges of robotics]. *IEEE Robotics & Automation Magazine* 14(1): 43–52.

Yu CH, Haller K, Ingber D and Nagpal R (2008) Morpho: A self-deformable modular robot inspired by cellular structure. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 3571–3578.

Acknowledgements

I would first like to thank my thesis supervisor Prof. Lorenzo Sabattini of the Automation, Robotics and System Control group at University of Modena and Reggio Emilia. The door to Prof. Sabattini office was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this dissertation to be my own work, but steered me in the right the direction whenever he thought I needed it.

I would also like to thank the experts who were involved in the validation survey for this research project: Prof. Roderich Groß, Dott. Ing. Andreagiovanni Reina, Dott. Ing. Yuri Kaszubowski Lopes. Without their passionate participation and input, the validation survey could not have been successfully conducted.

I would also like to thank my friends and university classmates, F. Benzi, A. Pupa, M. Fiorani, A. Masia that contributed to make these last years pleasant and fun. Thank you for all the hours spent and enjoyed on the different university projects, often also until late at night.

I would like also to thank the childhood friends from *ESPA* and *Sanpro* that in different ways, through thoughts, messages, laughs and chats always supported me in difficult moments.

Finally, I must express my very profound gratitude to my family, Angelica, Marco e Rossana, and in particular to my parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.