Graz University of Technology

Matteo Lasagni, Dott. Dott. mag.

Shape-Shifting Materials: from Mechanical Design to Dynamic Models

---

DISSERTATION

zur Erlangung des akademischen Grades
Doktor der Technischen Wissenschaften (Dr. techn.)

eingereicht an der
Technischen Universität Graz

Betreuer
Univ.-Prof. Dipl.-Inform. Dr. sc. ETH Kay Uwe Römer

Institut für Technische Informatik

Graz, im Mai 2017

EIDESSTATTLICHE ERKLÄRUNG

AFFIDAVIT

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Dissertation identisch.

I declare that I have authored this Thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present doctoral dissertation.

....

(Datum / Date) (Unterschrift / Signature)

Alla mia famiglia
To my family

Acknowledgments

I am not completely sure whether the flapping wings of a butterfly can remotely generate a hurricane, but I am very well aware that an unknown person met in San Francisco can change the flow of somebody's life in Europe!

This fantastic journey that began in the North of Germany and continued in the heart of Austria, gave me not only the opportunity to produce the scientific results contained in this dissertation, but especially it granted me the chance to meet awesome persons, reinforce lifetime friendships and establish new ones. All together, this has contributed to my personal and professional growth along a path that, despite the inevitable difficulties, has been absolutely worth to be walked and lived every day. Far from being two pages sufficient to express my gratitude to whom has contributed in many ways to this accomplishment, I seize the opportunity to thank everybody and, in particular, those persons who made this experience even more remarkable.

In the first place, I would like to thank Prof. Kay Römer for offering me the opportunity to pursue my Ph.D on a very innovative topic, perfectly matching my personal and professional interests, such as Shape-Shifting Material. The valuable remarks and feedback I received from him throughout the past years, not only have contributed to improve the quality of my scientific results, but also have had a positive impact on my mindset and personal skills. I am very grateful for his guidance and for the time he dedicated to discuss ideas and solutions, to meticulously correct my drafts (including this dissertation), and to suggest improvements for talks and presentations.

I also take the chance to thank Prof. Julien Bourgeois, who agreed on being the external examiner for my dissertation and whose feedback helped me improve this document.

One of the person who has played a significant role in all this journey, is Carlo to whom goes my deep gratitude at both professional and personal levels. He is the "butterfly effect" man I met one day in San Francisco and who linked me to Prof. Kay Römer. For this, I owe him a lot of this great experience that began after our meeting. Never would I have expected on that day on the West Coast, what a future was awaiting me just because of this fortuitous encounter. As the great friend he has become, he was the one welcoming me in Lübeck and with whom I have shared many adventures also here in Graz. His shoulder has been always ready to give me both technical and moral support through all the ups and downs that life, and especially a life as a doctoral student abroad, brings along.

In addition to Carlo, I am very grateful to all those friends with whom I spent memorable time in Lübeck and who contributed to make those cloudy days, characteristic of the North of Germany, shine (in particular Nóra, Silvia, Marta, and Dean). Great memories link me also to the colleagues in Lübeck (Felix, Richard, Cuong, Patrick) who, together with Kay, Carlo and myself, had interesting discussions at lunch time as well as

interminable contests at the table soccer.

The life in Graz has also brought new valuable persons into my life. Starting with thanking the numerous Italian (and not only Italians) friends I met through the "Italians in Graz" group, I also thank my colleagues at the institute and the members of my research group, including Nóra. Among them, Carlo and Felix stand out for the prompt help provided in many situations: from dealing with minor technical issues, to struggling with the Austrian bureaucracy; Bernhard and Marco have often been inclined to engage in interesting and useful scientific discussions on various topics. A great thanks goes to Marcel for his genuine friendship throughout these years and his positive presence in many situations. I would also like to mention Marcel's AES team for the contagious happiness they spread around at the institute every day.

Back to my hometown, sincere thanks go to my lifelong friends in Cavazzoli (Reggio Emilia). Every time I travel back to Italy, I can rely on their immutable friendship, exactly as if I had never left. Among all, I would like to thank Marco, Andrea B., Paolo, Elisabetta, Francesca, Susanna, Lara. A special thanks goes to Andrea M. whose daily "presence" kept my mood up and made me feel solidly connected to my origin. Many thanks also go to my friends from university and from high-school, for the pleasant time they often reserved for me whenever I was home for a couple of days.

Aware that I should cast this in stone, a sincere *thank you* goes to my girlfriend Manjusri for her love, her patience, and her constant priceless support. She has the ability to cheer me up whenever needed and has always been very supportive and caring "no-matter what it takes". One day, for example, after a long night I spent in the office struggling with a publication close to deadline, she spontaneously stood up in the very early morning to prepare and bring me a "survival kit" (food and beverage). Thank you, Manni: we both now I would have "stumbled" several times without your help.

And finally my deepest thanks goes to my family. They are the foundations that sustained me in all possible ways while undertaking this wonderful experience. I dedicate these words and this thesis to my parents, my brother, my grandmothers, my grandfather and Bruna. They have encouraged me countless times, always telling me to never give up and to keep going towards my goals and expectations.
My heartfelt *thank you* to all of you!

Graz, May 2017
Matteo Lasagni

Ringraziamenti

Non sono completamente sicuro se il battito d'ali di una farfalla possa generare un uragano da qualche parte nel mondo, ma sono ben consapevole che uno sconosciuto incontrato a San Francisco può cambiare il flusso di vita di qualcuno che vive in Europa!

Questa fantastica esperienza iniziata nel nord della Germania e proseguita nel cuore dell'Austria, mi ha dato non solo l'opportunità di produrre i risultati scientifici contenuti in questa tesi, ma soprattutto mi ha permesso di incontrare molte persone, di rafforzare vecchie amicizie e di iniziarne di nuove. Tutto questo ha contribuito alla mia crescita personale e professionale attraverso un percorso che, nonostante le inevitabili difficoltà, è valso sicuramente la pena aver intrapreso e vissuto ogni giorno. Lungi dall'essere sufficienti due pagine per esprimere la mia gratitudine a chi ha contribuito in vari modi a questo mio risultato, colgo l'occasione per ringraziare tutti e, in particolare, quelle persone che hanno reso ancor più significativa questa esperienza.

In primo luogo vorrei ringraziare il prof. Kay Römer per avermi offerto l'opportunità di conseguire un dottorato di ricerca su di un argomento molto innovativo, perfettamente adattato ai miei interessi personali e professionali, come lo studio di materiali cambia forma. Le osservazioni e i suggerimenti che ho ricevuto da lui in questi anni hanno contribuito non solo a migliorare la qualità dei miei risultati scientifici, ma hanno anche avuto un impatto positivo sul mio modo di affrontare nuovi problemi così come sulle mie capacità. Sono molto grato per la sua guida e per il tempo dedicato a discutere idee e soluzioni, a correggere meticolosamente i miei scritti (compresa questa dissertazione) e per i suggerimenti volti a migliorare le mie presentazioni.

Colgo l'occasione per ringraziare il prof. Julien Bourgeois per aver accettato il ruolo di esaminatore esterno e per i suggerimenti ricevuti.

Una persona che ha svolto un ruolo significativo in questa mia esperienza è Carlo, a cui va la mia profonda gratitudine sia a livello professionale che personale. È lui la persona dell'effetto "farfalla", che ho incontrato un giorno a San Francisco e che mi ha messo in contatto con il prof. Kay Römer. Per questo gli devo buona parte di questa grande esperienza che ha avuto inizio dopo il nostro incontro. Non avrei mai immaginato quel giorno sulla costa occidentale, quale futuro fosse lì ad attendermi a causa di questo incontro fortuito. Da grande amico quale è diventato, è stato lui ad accogliermi a Lubecca e con il quale ho condiviso molte avventure anche qui a Graz. La sua spalla è sempre stata pronta a darmi sia supporto tecnico che morale durante tutti gli alti e bassi che la vita, e soprattutto la vita di dottorando all'estero, porta con sè .

Oltre a Carlo, sono molto grato a tutti quegli amici con i quali ho trascorso momenti memorabili a Lubecca e i quali hanno contribuito a rendere splendenti quei giorni nuvolosi, caratteristici della Germania del nord (in particolare Nòra, Silvia, Marta e Dean). Bei

ricordi mi legano anche ai colleghi di Lubecca (Felix, Richard, Cuong, Patrick) con i quali, assieme a Kay e Carlo ho avuto interessanti discussioni all'ora di pranzo e partite interminabili al calcio balilla.

Il trasferimento a Graz ha portato nuove persone importante nella mia vita. Comincio con il ringraziare i numerosi amici italiani (e non solo italiani) che ho incontrato attraverso il gruppo "Italiani a Graz". Ringrazio anche i colleghi dell'istituto e i membri del mio gruppo di ricerca, inclusa Nóra. Tra questi, Carlo e Felix si distinguono per l'aiuto fornitomi in svariate situazioni: dal trattare piccoli problemi tecnici, al gestire la burocrazia austriaca; Bernhard e Marco sono stati spesso disposti a intraprendere interessanti e utili discussioni scientifiche su svariati argomenti. Un grande grazie va a Marcel per la sua genuina amicizia durante questi anni e per la sua presenza positiva. Inoltre, vorrei menzionare il gruppo di ricerca di Marcel (AES) per la felicità contagiosa che diffonde continuamente in ufficio.

Tornando alla mia città natale, un grazie sincero va a i miei amici di sempre di Cavazzoli (Reggio Emilia). Ad ogni mio rientro in Italia, posso contare sulla loro immutabile amicizia, esattamente come se non me ne fossi mai andato. Anche se vorrei citare ogni singolo nome, tra tutti ringrazio Marco, Andrea B., Paolo, Elisabetta, Francesca, Susanna, Lara. Un ringraziamento speciale va ad Andrea M. la cui "presenza" quotidiana ha mantenuto alto il mio stato d'animo e mi ha fatto sentire saldamente legato alla mie origini. Il mio grazie va anche agli amici dell'università e della scuola superiore, per il tempo piacevole che spesso hanno trovato per me ad ogni mio rientro.

Consapevole che dovrei incidere questo su pietra, un sincero grazie va alla mia fidanzata Manjusri per il suo amore, la sua pazienza e il suo costante ed inestimabile supporto. Lei è quella che fa il tifo per me ogni volta che è necessario ed è sempre disponibile e premurosa "costi quel che costi". Un giorno, ad esempio, dopo una lunga notte che ho trascorso in ufficio per concludere una pubblicazione, lei di sua spontanea iniziativa si è alzata la mattina presto per prepararmi e portarmi un "kit di sopravvivenza" (cibo e bevande). Grazie, Manni: sappiamo entrambi che sarei "inciampato" diverse volte senza il tuo aiuto.

E infine il mio ringraziamento più profondo va alla mia famiglia. Loro sono le fondamenta che mi hanno sostenuto in tutti i modi possibili durante questa meravigliosa esperienza. Dedico queste parole e questa tesi ai miei genitori, a mio fratello, alle mie nonne, a mio nonno e a Bruna. Loro mi hanno incoraggiato innumerevoli volte, dicendomi sempre di non lasciare perdere e proseguire verso i miei obiettivi e le mie aspettative. Grazie di cuore a tutti voi!

Graz, Maggio 2017
Matteo Lasagni

Abstract

Shape-Shifting Materials will revolutionize the way we conceive and interact with things around us. Novel materials with the unprecedented ability to change their physical properties on demand can be built upon the combination of a massive number of tiny *programmable robotic particles*. This means that with the same ease we nowadays modify the software in our computers, in the future we will program the physical features of things. Novel applications as well as new human-matter interaction paradigms emerge in this perspective. For example, *shape-shifting displays* show *tangible* 3D scenes, which users can directly interact with. On a larger scale, "editable" environments promptly rearrange furniture and walls to turn, e.g., an office into a party room. As materials can be programmed to self-repair and to self-reproduce in the same way living organisms self-heal and self-replicate, maintenance-free products will eliminate manufacturing costs.

This work focuses on the mechanisms (hardware and software) necessary to realize Shape-Shifting Materials. To deeply investigate this, we concentrate on the realization of a *tangible shape-shifting display*, which is a mechatronic device designed to render *arbitrary shapes* at a *high resolution*. Several design challenges and conflicting requirements arise at different abstraction levels. A major challenge concerns the number and the dimensions of the robotic particles. While particle dimensions decrease for higher resolution, the number of particles needs to increase in order to obtain reasonably sized shapes. This entails a scalability challenge, as actuators and latches embedded in particles – to enable shape shift – need to withstand a growing number of particles, whereas their actuation forces diminish with their size. Also, as the complexity of particles increases along with their freedom to dynamically rearrange, the integration of sophisticated *mechatronic* mechanisms hampers particles miniaturization.

Our thesis is, in the first place, that the formation of arbitrarily complex shapes does not necessarily imply sophisticated mechanisms, rather that particle freedom of movement can be traded for higher system scalability. The presented solution overcomes the inverse relationship between miniaturization and strength of actuators through the introduction of simple built-in mechanisms that, instead of actively actuating a particle, "guide" an externally provided force to actuate the particle. In this way, the external force can increase to actuate a growing number of particles and thus the system is scalable. Also, the minimalistic mechanical design makes particles amenable for miniaturization. Not only is scalability a challenge at the mechanical level, it also becomes an issue at the control level. Inappropriate control strategies might be unable to effectively actuate the system, in this way limiting its scalability. In the worst case, too intense actuation forces, even if only temporary, undermine the stability and the integrity of the whole system. To prevent such a situation, static and dynamic models are required to enable optimal

planning and control and to predict the effects of actuation on the system. Considering the potentially large number of mechanical elements and the fact that our specific design is underactuated, the formulation of such models is non-trivial.

Our contribution is twofold. First, we identify and solve system scalability and particle miniaturization issues at the mechatronic level. This leads us to the design and the realization of prototypes that, through a progressive refinement process, eventually overcome the limitations described above. Second, we derive static and dynamic models to predict the response of the system undergoing a shape-shift. This is necessary to enable optimal planning and control algorithms and by that maximize the overall scalability of the system towards high resolution rendering.

Zusammenfassung

Formveränderliche Materialien werden die Art und Weise revolutionieren, in der wir die Welt um uns herum wahrnehmen und mit ihr interagieren. Aus *kleinen programmierbaren Teilchen* können neuartige Materialien mit der einmaligen Fähigkeit geschaffen werden, ihre physikalischen Eigenschaften je nach Anforderung kontrolliert zu verändern, genauso wie wir heutzutage die Software von unseren Computern ändern können. Hierdurch entstehen gänzlich neue Anwendungen und Interaktionsmöglichkeiten zwischen Mensch und Materie. Zum Beispiel können *Anzeigen mit veränderlicher Oberflächenstruktur fühlbare Szenen in 3D* darstellen und den Nutzer direkt interagieren lassen. In größerem Maßstab können Möbelstücke und Wände von Räumen einfach angepasst und verändert werden, um z.B. ein Büro in einen Partyraum zu verwandeln. Darüber hinaus könnten Materialien so angelegt werden, dass sie sich selbst reparieren und vervielfältigen. Dadurch können wartungsfreie Produkte hergestellt und bei Bedarf beim Kunden kostengünstig erzeugt werden.

Das Hauptaugenmerkt dieser Arbeit liegt auf den Mechanismen (Hardware und Software) zur Formmanipulation von Objekten. Um dieses Konzept tiefgehend zu untersuchen, konzentrieren wir uns in dieser Arbeit auf die Umsetzung einer *formveränderlichen Anzeige*. Dieses mechatronische Gerät kann beliebige *Formen mit hoher Auflösung sichtbar und fühlbar* machen. Eine der größten Herausforderungen im Design betrifft die Anzahl und Größe der mechatronischen Partikel. Während sich die Größe zugunsten einer höheren Auflösung verringert, muss die Anzahl der Teilchen für eine ausreichende Gesamtgröße der Anzeige zunehmen. Somit stellt die Skalierbarkeit ein weiteres Problem dar: Stellglieder und Verbindungen in den Partikeln müssen bei zunehmender Teilchenzahl wachsenden Belastungen standhalten, während ihre eigenen mechanischen Kräfte mit ihrer Größe abnehmen. Des Weiteren nimmt die Komplexität der Teilchen mit ihren Freiheitsgraden zu, wohingegen sich die dafür erforderliche Komplexität der mechatronischen Elemente erhöht und die Miniaturisierung erschwert.

Unsere erste zentrale These ist, dass die Erzeugung beliebig komplexer Formen nicht unbedingt eine ausgefeilte Mechanik erfordert, sondern dass der Freiheitsgrad der beweglichen Partikel im Hinblick auf eine höhere Skalierbarkeit des Gesamtsystems optimiert werden kann. Die vorgelegte Lösung zeigt, dass die inverse Beziehung zwischen Miniaturisierung und Kraft der Stellglieder durch die Integration einer einfachen Mechanik in jedes Teilchen überwunden werden kann. Anstatt Partikel aktiv zu bewegen, können sich diese der Kraft eines externen Aktuators bedienen. Diese äußere Kraft kann bei zunehmender Anzahl an Partikeln zugunsten der Skalierbarkeit gezielt erhöht werden. Zudem erleichtert das vorgestellte mechanische Design die Miniaturisierung der Partikel. Auch hinsichtlich der Regelung der Formveränderung ist Skalierbarkeit eine Herausforderung. Ungeeignete

Regelungsstrategien können die Bewegung und Skalierbarkeit der Partikel beeinträchtigen. In einigen Fällen können zu hohe Kräfte tolerierbare Grenzen überschreiten und damit die Stabilität des Gesamtsystems nachhaltig schwächen. Um solche Situationen zu vermeiden, sind statische und dynamische Modelle zur optimalen Planung und Regelung der Bewegungen bezüglich ihrer Auswirkungen notwendig. Aufgrund der potenziell großen Anzahl mechanischer Elemente und der Tatsache, dass unser spezifisches Design unteraktuiert ist, ist die Formulierung solcher Modelle nicht trivial.

Unser Beitrag ist zweierlei: Zunächst identifizieren und lösen wir Probleme der Skalierbarkeit und Miniaturisierung der Partikel auf mechatronischer Ebene. Dies ermöglicht uns die Realisierung von Prototypen, welche die genannten Einschränkungen überwinden. Zweitens leiten wir statische und dynamische Modelle ab, um die Reaktion der Systeme während einer Formänderung vorherzusagen.

Riassunto

I Materiali Cambia Forma rivoluzioneranno il modo di concepire e interagire con gli oggetti che ci circondano. Materiali innovativi, capaci di cambiare le proprie proprietà fisiche in maniera controllata, possono essere costruiti combinando un gran numero di *piccole particelle robotiche programmabili*. Ne deriva che con la stessa facilità con cui oggi modifichiamo il software nei nostri computer, in futuro cambieremo le caratteristiche fisiche degli oggetti. In questa prospettiva emergono nuove applicazioni e nuovi paradigmi di interazione uomo-materia. Ad esempio, *display cambia forma* mostrano *scene 3D tangibili*, con cui gli utenti possono interagire. Su larga scala, ambienti "editabili" trasformano tempestivamente mobili e pareti per convertire, ad esempio, un ufficio in una sala per ricevimenti. Dal momento che tali materiali possono essere programmati per auto-ripararsi e auto-replicarsi, nuovi prodotti a manutenzione zero eliminano anche i costi di produzione.

Questo lavoro di ricerca esplora i meccanismi (hardware e software) necessari per realizzare Materiali Cambia Forma. A tal fine, consideriamo la realizzazione di un *display cambia forma*, ovvero un dispositivo meccatronico che genera *forme 3D tangibili ad alta risoluzione*. L'analisi dei requisiti a diversi livelli di astrazione mostra spesso conflitti. Uno di questi riguarda il numero e le dimensioni delle particelle robotiche. Mentre le dimensioni delle particelle diminuiscono per aumentare la risoluzione del display, il loro numero deve aumentare per ottenere forme di dimensione ragionevole. Questo comporta un problema di scalabilità: gli attuatori integrati nelle particelle devono far fronte ad un numero crescente di queste ultime, mentre le forze di azionamento diminuiscono con le dimensioni. Inoltre, aumentando la libertà di movimento delle particelle, aumenta la loro complessità meccanica che ostacola la possibilità di miniaturizzarle.

La tesi qua presentata è, in primo luogo, che la generazione di forme complesse non richiede meccanismi sofisticati. Al contrario, la libertà di movimento delle particelle può essere ridotta per ottenere una maggiore scalabilità del sistema. Il problema dovuto alla relazione inversa tra miniaturizzazione e forza degli attuatori è risolto attraverso semplici meccanismi che, invece di attuare attivamente una particella, "guidano" una forza esterna al fine di azionare la particella. In questo modo, la forza esterna può essere aumentata per gestire un numero crescente di particelle e quindi il sistema è scalabile. Inoltre, la semplicità del meccanismo facilita la miniaturizzazione delle particelle.
La scalabilità è non solo una sfida a livello meccanico, ma anche un problema di controllo. Strategie di controllo inadeguate compromettono l'attuazione del sistema, limitandone la sua scalabilità. In casi estremi, forze di attuazione troppo intense, anche se solo temporanee, minano la stabilità e l'integrità dell'intero sistema. Per evitare tale situazione, sono necessari modelli statici e dinamici che consentano una pianificazione e un controllo ottimali e che prevedano gli effetti dell'attuazione sul sistema. Considerando il numero

potenzialmente elevato di elementi e il fatto che il sistema presentato è sottoattuato, la derivazione di tali modelli è difficoltosa.

Il nostro contributo è duplice. Innanzitutto, individuiamo e risolviamo i problemi di scalabilità del sistema e di miniaturizzazione delle particelle a livello meccatronico. Questo ci porta alla progettazione e alla realizzazione di prototipi che, attraverso un progressivo processo di miglioramento, superano le limitazioni sopra descritte. In secondo luogo, deriviamo modelli statici e dinamici per prevedere la risposta del sistema durante un cambiamento di forma. Questo è necessario per la creazione di algoritmi di pianificazione e controllo ottimali e, di conseguenza, per massimizzare la scalabilità complessiva del sistema e della sua risoluzione.

Contents

List of Figures

List of Tables

List of Abbreviations

AC Alternating Current

CAD Computer-Aided Design

COA Chain-Oriented Approach

DC Direct Current

DNA Deoxyribonucleic Acid

DOF Degree of Freedom

GUI Graphical User Interface

HD High-Definition

HW Hardware

IoT Internet of Things

LCD Liquid-Crystal Display

LED Light-Emitting Diode

MEMS Micro-Electro-Mechanical Systems

NE Newton-Euler

NEMS Nano-Electro-Mechanical Systems

PM Programmable Matter

POA Particle-Oriented Approach

SW Software

TUI Tangible User Interface

UI User Interface

WSN Wireless Sensor Network

SMA Shape-Memory Alloy

SSD Shape-Shifting Display

SSM Shape-Shifting Material

Chapter 1

Introduction

Programmable Matter (PM) indicates a novel class of materials with the unprecedented ability to change their physical properties on demand, in a controlled manner. The term, coined in 1991 by Toffoli and Margolus [106] to indicate a 3D-array of computational cells, has been redefined in 2004 by Goldstein et al [44] to refer to novel materials composed of modular robotic elements. The key concept is that, by controlling the collective behavior of these robotic elements, innovative structures with adaptable properties can be obtained. This potentially leads to the creation of materials able to self-heal, self-adapt, and even self-reproduce in a similar way to living organisms.

The idea of creating machines able to emulate characteristics of the living organisms, dates back to 1959 when Penrose [83] proposed a self-reproducing mechanical device. Later, in 1988, Fukuda et al. [39] devised modular robotic systems able to dynamically self-reconfigure in order to adapt themselves to challenging environments and to execute tasks which are unpredictable at design time. Originally intended to support critical missions [126] (e.g., rescue and space missions where limited resources call for high adaptability) *modular robotics* has become an established research field [58]. This lays the foundation for the creation of PM built upon the aggregation of *modular robotic particles*, where each programmable particle controls the local features of the material.

Science-fiction has many times contributed to depict the idea and the potential of PM. For example, in the movie "Terminator 2" (1991), the actor Robert Patrick plays the role of "Terminator T-1000", a shape-shifting android able to modify his body phase from solid to liquid. This allows him to teleport his body anywhere, to take on the appearance of anybody, as well as to instantly grow new features like swords in place of hands. More recently, in 2006, the Disney movie "Big Hero 6" pictures a *nano-robotic particle system* that can adjust its shape, self-replicate, pass through narrow pipes, create walls and bridges on demand. All this is *ideally* carried out by controlling the interaction among the smart particles composing the system.

Programmable Matter (PM) has the potential to revolutionize our daily life and the way we interact with things around us. For example, instead of buying a new couch, a PM couch could update its consistency, its color and even its shape to match our new desires. Similarly, the smart particles of a broken PM jar can autonomously reassemble in the same way that the cells of an organic tissue self-heal. In this perspective, significant economical and social impacts are to be expected. For example, "green" products with unlimited lifetime, could self-recycle and transform into new products, thereby avoiding tons of

waste. Design and manufacturing approaches would also radically evolve. As we can remotely control the features of an object, instead of manufacturing and delivering goods, in a future scenario products could be "sent" through the Internet and manufactured directly at customer's destination.

Particularly interested in shape manipulation, we focus in this work on the realization of "Shape-Shifting Materials". Through this term we emphasize the ability of PM to create *arbitrary* shapes by spatially rearranging the robotic particles of its structure.
Our main quest concerns the realization of *high-resolution* Shape-Shifting Materials (SSMs), for which scalability – in terms of size and number of particles – is a fundamental requirement. Main challenges concern the design and the manufacturing of a scalable system as well as the coordination of *potentially* thousands of elements.

We argue that scalability is primarily a mechanical design challenge. Our solution grounds on the observation that the formation of arbitrarily complex shapes does not necessarily require sophisticated mechanisms embedded in the robotic particles, which would otherwise compromise particle miniaturization, system scalability, and cost-effectiveness. We notice that even when solved at the hardware level, scalability remains a challenge at the software level for the coordination of the ensemble of particles during a shape shift. Inappropriate shape-shifting approaches might indeed cause intense actuation forces to undermine the mechanical integrity and stability of the whole system.

In this chapter, we present some potential applications based on SSM and more in general on PM. We state the major challenges towards the realization of SSM and elaborate potential solutions. To help the reader better understand the following chapters, we introduce our methodology and contributions. A concrete case-study is also defined to effectively drive our quest thorough this work. In the following, we sometimes use the terms SSM and PM interchangeably, aware of the fact that PM includes SSM.

1.1 Shape-Shifting Material: Potential and Challenges

Shape-Shifting Materials (SSMs) open a wide range of innovative applications and paradigms. As the spatial arrangement of the modular programmable particles forming an SSM can be software controlled to make new shapes emerge, new functionality can be "installed" into objects made of SSM. For example, an additional drawer can be installed in an SSM desk to better organize our documents, with the same ease we add a new folder to the file system of our computers. In this perspective, not only will the way we manipulate and interact with things around us change, but also new trends should be expected regarding the development, manufacturing and maintenance of conventional products. In the following, we consider some potential SSM applications.

Instant-prototyping. SSMs facilitate the development of new products [7,43] by enabling "instant-prototyping", as the next step beyond "fast-prototyping" today possible with the advent of 3D-printing technologies. Despite the availability of advanced simulators and Computer-Aided Design (CAD) software to model and predict the behavior of arbitrarily complex mechanisms, the realization of prototypes remains a necessity to experimentally evaluate a new product before its release on the market. While conventional prototyping undergoes the same manufacturing process of normal products, with significant waste of resources and time, newly available technologies, such as 3D printing, enable fast-prototyping to significantly improve the whole process. With SSMs, a further technological leap becomes possible, as shapeless bunch of matter could instantaneously "materialize" any virtual model dwelling in the memory of a computer (e.g., CAD model), thereby drastically reducing the time-to-prototype. Furthermore, the possibility to endlessly re-program the same SSM into new shapes minimizes the waste of resources.

Cyber-physical duality. Programmable Matter (PM) and SSM establish a *mapping* between the *cyberspace* and the *physical world* [14, 55, 56, 71]. Indeed, a *virtual model* and its materialization into a *physical prototype* are representations of the same entity. If we could exploit such a mapping in both directions, namely to feed any change made to the physical world (i.e., the prototype) back into the cyberspace (i.e., update the model), new design paradigms would emerge. Conventional *data-centric* design (i.e., design based on technical drawing or CAD models) would leave room to more innovative *proto-centric* design, where the dexterity of the designer to "mold" new ideas directly into the matter, outperforms traditional approaches based on computer graphics. A similar vision dates back to 1997, when Ishii et al. [56] introduce the concept of "tangible bits" which could potentially transform traditional Graphical User Interfaces (GUIs) into Tangible User Interfaces (TUIs). This prospects direct *human-matter interactions*, where *hand-touch* and *gestures* encode commands to control the shape and any other feature of PM (Ishii et al. [55]). For example, through *hand-touch* users can "mold" the outer shape of an object, as if it were made of clay, while with *gestures* users can mimic the movement of a paint brush modifying the color of its surface.

Shape-Shifting Display. Novel human-*matter* interaction paradigms have the potential to even reinforce existing human-*human* interaction models, by introducing new means to communicate and to convey ideas. For example, the realistic 3D vision that innovative Shape-Shifting Displays can provide, enables teams of heterogeneous domain experts to easily share complex ideas and visualize advanced concepts. Also, through TUIs [56] users can interact with the displayed objects by simply swiping their finger

tips on the surface of the display, as we do nowadays with our smartphones. This allows users to zoom in on specific details, rotate an object in the three dimensions, and even "disassemble" the object in order to visualize its internal structure.

Towards this scenario, Jansen et al. [57] introduce the concept of "Data Physicalization" as further step beyond data visualization, and discuss the numerous opportunities that tangible information can bring. In [36], Follmer et al. propose the design of a programmable surface, "inFORM", which is able to modify its shape and colors as well as to react to users' hand-gestures. Beside allowing the visualization of 3D shapes (e.g., 3D histograms), inFORM enables the remote manipulation of objects lying on its programmable surface.

3D-fax and 3D-telephone. The interaction and the cooperation among people might also benefit of visionary "3D-fax machine" [86] and 3D telephones (telepario [85]). PM can not only be used to create new objects but also to 3D-scan shapes and colors of existing ones, for example, by applying a layer of PM on their surfaces. In this way, the 3D model of an object can be transmitted through the Internet and eventually materialized at a remote location. In a less invasive way, the 3D model of a person can be acquired by means of a 3D scanner and shown on a remote shape-shifting display; the rapid repetition of this process enables futuristic 3D-telephone calls.

The use of 3D-fax might also facilitate remote maintenance operations. While a new life-cycle should be expected for next-generation products made of PM (which potentially become maintenance-free), replacement parts for more conventional items could be "3D-faxed" instantaneously through the Internet.

Industry 4.0. The fourth industrial revolution "Industry 4.0" [101] aims at high standards of automation towards the creation of the so-called "smart factory". Autonomous cyber-physical systems monitor the manufacturing process and make decisions, for example, in order to optimize the configuration of production lines to match the current necessities. In this vision, where fusion and cooperation between humans and cyber-physical systems is maximized, SSMs enable a large variety of new scenarios. For example, "shape-shifting workbenches" enable tools and physical supports to adapt to the product being assembled, in this way facilitating the assembly operations; "programmable molds" can promptly adjust their shapes to allow massive production of highly customized products, which are perfectly tailored to customer's desires; robotic manipulators can be generated on demand to be optimally configured for the current set of tasks.

Reconfigurable environment. Applied to a larger scale, Weller et al. [108] envision reconfigurable environments, where furniture and walls made of SSM can easily adapt to various situations. In this perspective, an office can quickly transform into a dining room and, in case of unexpected guests, chairs and tables emerge from the floor.

1.1.1 Challenges.

The realization of PM and SSMs raise many challenges at both hardware and software levels. These challenges concern both the design of an individual *particle* and of the *system* as an ensemble of particles. Table 1.1 highlights the main challenges at each intersection of Hardware (HW) and Software (SW) with *particle* and *ensemble*.

From a hardware perspective, a particle should be a cost-effective *miniaturized* device that autonomously actuate and interconnect to other particles through physical interfaces.

	Particle	Ensemble
HW	• *miniaturized* size • cost-effective • autonomous actuation • interface to other particles	• *scalable* • mechanical robustness • power supply • networking
SW	• *control* local actuation • interact with neighbors • communicate with neighbors • detect local failures	• coordinate shape shift • *planning* algorithms • communication protocols • failure recovery

Table 1.1: Main challenges at the HW and SW level, seen from particle and ensemble perspectives.

The latter have to ensure mechanical stability and also to provide a medium for supplying power to other particles and for networking. From a software perspective, a particle should be able to control its local actuation, to communicate, to interact with other particles, and to detect local failures.

From a hardware perspective, the ensemble of particles need to be *scalable* – in terms of number of particles – and mechanically robust. From a software perspective, to efficiently *coordinate* the ensemble of particles, planning algorithms and communication protocols are required. Also, failures of single particles or groups of particles need to be handled.

Among the potential challenges, we address in this dissertation the following problems from both hardware and software perspectives. The first problem we tackle, concerns the design of a *scalable* system of *miniaturized* particles. The challenge lies in the fact that the actuators embedded in particles need to be sufficiently strong to support a growing number of particles, while their dimensions need to be sufficiently small to facilitate particle miniaturization. As in general strength and dimensions of actuators are correlated, a scalability challenge exists.

The second problem that we consider, concerns the derivation of static and dynamic models necessary to predict the behavior of the system when actuated and thereby to enable optimal *planning* and *control* strategies. Major challenges lie in the fact that these models need to be both *accurate* and *computationally efficient*, despite the large number of mechanical elements to be taken into account. Also, as the prototypes presented in Chap. 3 combine rigid and compliant elements, the derivation of dynamic models is non-trivial and only partially covered by the current state of the art.

1.2 Problem Statement

In our vision, a Shape-Shifting Material (SSM) is the combination of many mechatronics devices (i.e., programmable particles) forming a dense topology. A network connects such devices with the dual goal of supplying power and enabling communication among particles that can in this way cooperate to collectively modify the morphology of the shape

they form. To orchestrate such a complex system, shape-shifting algorithms define how particles re-arrange and schedule their actuation with the objective of maintaining the system integral and stable.

Although the ideal scenario portrays a universal SSM suitable for any application and able to take on any shape, the realization of an SSM is a difficult task because of the many inter-disciplinary intertwined challenges and conflicting requirements that rise at different abstraction levels. This complicates not only the mechatronic design of the elementary particles, but also makes the actuation of the whole SSM a non-trivial problem. While on the one hand lightweight and simple mechanical designs are preferable, on the other hand too minimalistic solutions might reduce the controllability of the system. For example, one of the most elementary form of programmable matter are self-assembly structures that emerge from the random interaction of tile particles floating in a fluid [51]. In these systems, the assembly rules governing shape formation are encoded in the geometry of the particles, whose edges outline interlocking patterns similar to the pieces of a jigsaw puzzle. Shape formation is therefore an aleatory process, triggered by the turbulence of the fluid in which particles are suspended. Even though the absence of actuators embedded in particles results in a lightweight and simple mechanical design, whether and when a target shape will eventually emerge is hard to predict and to control. This makes these self-assembly systems unsuitable for many applications.

By analyzing the relationship among requirements at different abstraction levels, we identify the conflicts that make the design of an SSM particularly challenging. This allows us to act on the causes and devise solutions to avoid conflicts, or at least to contain their negative effects.

1.2.1 Particle Topology and Shape Formation

One of the first and foremost questions about the design of the elementary programmable particles concerns the mobility of particles to modify their spatial arrangement, and thus to form new shapes. Generally, the higher the freedom of particles to re-arrange the more complex is their architecture. Thus, in order to allow the formation of arbitrary shapes but also to contain costs, a trade-off between particles' freedom of movement and mechanical complexity must be found.

We distinguish between two types of architectures, mainly characterized by the topology that the robotic particles form: *dynamic*- and *static*-topology architectures. In the first case, particles can temporarily detach from the ensemble and migrate to a different location, for example by climbing each other, and by that create convex or concave geometries to best approximate the target shape. The freedom of particles forming such "dynamic" or "detachable" topologies, resemble the freedom of molecules of a liquid that, free to flow relatively to each other, can seamlessly adapt to any container. In contrast, particles forming "static" or "non-detachable" topologies are constrained to a predefined neighborhood, where only relative displacements between adjacent particles, or particle deformations, are allowed in order to carry out a shape-shift.

While on the one hand, detachable topologies intuitively allow the formation of literally any shape, on the other hand, particle detachment and relocation call for sophisticated mechatronic designs that hamper particle miniaturization and lead to higher costs. Also, as power need to be externally supplied (in order to avoid bulky batteries embedded in

particle), temporary power supply interruption need to be handled during particle relocation. Furthermore, as the network topology changes over time, communication among detachable particles requires protocols to handle mobile routing and possible data loss.

Non-detachable topologies have the advantage that (1.) mechanisms can be relatively simple, (2.) power supply and networking are not subject to frequent disconnections, and (3.) communication among particles can rely on a predefined consistent topology. However, because these architectures allow only relative displacements among adjoining particles, the set of admissible shapes is limited. This leads to our research question:"To which extent does particle topology affect shape formation?"

1.2.2 Scalability and Particle Miniaturization

The inherent relationship between the maximum forces and the physical size of actuators and latching mechanisms, entails an upper-bound to system scalability and a lower-bound to particle miniaturization. Aiming at the creation of a high resolution rendering SSM, while the dimensions of particles tend to diminish, their number needs to increase in order to allow reasonably sized shapes. This calls for stronger actuators and latching mechanisms to withstand a growing number of particles. However, along with their dimensions also the forces that embedded actuators and latches can exert, diminish. This causes a scalability issue.

One can argue that while the dimensions of particles decrease, their "density" (i.e., number of particles per cubic unit) increases; thus, the overall intensity of actuation and adhesion forces does not change because of the complementary contributions of multiple particles [69]. Despite this observation being true for the general case, scalability constraints yet remain when aiming at rendering thin geometries. For example, configurations consisting of one single particle (e.g., a long beam or a skeleton) or one single layer of particles (e.g., a wide thin surface) tend to easily break apart [114]. This is due to the insufficient number of bonds that exist at some cross-sections of the target geometry and which are not strong enough to counteract the external forces tearing the SSM apart.

The above relationship between force and size of actuators implies a lower-bound also to particle miniaturization. As in general the actuators embedded in particle should guarantee a minimal force sufficient to actuate the particle, their size cannot be arbitrarily reduced nor can the particle be arbitrarily scaled down.
In addition, considering that a shape-shift results from the spatial rearrangement of particles, the potentially complex mechanisms necessary to make this possible, hamper particle miniaturization. Mechanical solutions should be found to guarantee system scalability and particle miniaturization.

Our research question in this perspective is: "Which design principles can avoid, or at least minimize, the limitations that derive from the inherent relationship between force and size of embedded actuators and latches? Which design principles maximize scalability and make particles amenable for miniaturization?"

1.2.3 Planning and Control

Although scalability primarily poses a mechanical design challenge, it also requires appropriate shape-shifting strategies to prevent excessive actuation forces from damaging the

system. Even if only temporarily applied, too intense actuation forces might derive from inappropriate configurations of the system shifting from an initial to a target shape.

Starting from an initial shape, a shape-shift results from the collective actuation of particles, which spatially re-arrange into the target shape. Given the large number of elements to control, a shape-shift is a process that generally requires multiple intermediate configurations, whereby the SSM gradually transforms towards the target shape. Global *planning* is required to solve the decision problem about which particles to actuate at each step, namely which sequence of intermediate configurations to select. Planning algorithms need to take into account two types of constraints: the physical constraints limiting particles reciprocal movements, and the physical constraints that need to be satisfied at each point in time to keep the system integral.

A target shape is said to be *feasible* if (1.) the physical constraints of the system do not prevent particles from arranging into the target shape; *and* if (2.) the forces necessary to retain the target shape do not exceed the maximum limit; *and* if (3.) the forces necessary to carry out the corresponding shape shift do never overcome the maximum force that the system can withstand. Conditions (1.) and (2.) are *static* conditions that only depend on the system design and on the target shape. The third condition (3.) is a *dynamic* condition that depends on the shape shifting approach (i.e., on the planning algorithms). In case (1.) and (2.) are not satisfied, the target shape is certainly *not* feasible. In case only (3.) is not satisfied, a different shape shifting approach might solve the problem.

When a proper planning is found, *control* algorithms coordinate the actuation of the system. In order to enable optimal planning and control, *static* and *dynamic* models are required to asses the feasibility of the target shape (1. and 2.) and to predict the behavior of the system (3.) during the transient between consecutive intermediate configurations. Considering the large number of mechanical elements combined together, the derivation of such models is quite challenging.

Our quest aims at finding a possible trade-off between accurate and computationally efficient models to support planning and control algorithms.

1.3 Thesis Statement

This doctoral thesis demonstrates that the formation of arbitrarily complex shapes does not necessarily require sophisticated *mechatronic* mechanisms to enable particle spatial rearrangement, and furthermore that particle *freedom of movement* can be traded for *higher system scalability* and *particle miniaturization*.

In this work, we demonstrate that scalability is not only a mechatronic design challenge, but it also depends on the strategies adopted to coordinate the actuation of the system. In particular, we argue that wrong actuation strategies compromise system's stability and integrity, hence limiting its scalability. Optimal planning and control algorithms are required to resolve, or at least to mitigate, this issue. We show that static and dynamic models can be combined to accurately and efficiently predict the behavior of the system being actuated and to support optimal planning and control algorithms.

1.4 Contributions

The scientific contributions of this dissertation are summarized as follows:

- The ultimate goal of this work is the realization of a shape-shifting display, a mechanical device able to approximate the outer surface of 3D objects. A target key-feature is the formation of arbitrary shapes at high-resolution rendering, for which many tiny modular robotic particles are combined. This requires the system to be *scalable* to support a growing number of particles, and particles to be *miniaturizable* to allow high-resolution rendering.

  We observe that existing state-of-the-art architectures hamper system scalability and particle miniaturization, because of the complex mechanisms which are embedded in particles. In particular, the maximum strength of actuators limits the scalability of the system, while their dimensions are hardly miniaturizable.

  To overcome these limitations, our first contribution consists in the definition of a novel design principle that we call "Force-Guiding Principle". We devise a solution to remotely actuate particles by means of an external actuator. In this way, the external actuator can be upgraded to support a growing number of particles, while particle can be miniaturized.

  To verify the applicability of the "Force-Guiding Principle", we design and realize a system based on chains (non-detachable topology), where only relative displacements among adjoining particles are allowed. We also demonstrate that despite the physical constrains among particles, the system can take on arbitrary shapes.

- Our second contribution is the derivation of static and dynamic models to predict the behavior of the system when actuated. This is fundamental to support planning and control algorithms, necessary to coordinate the actuation of the system and to guarantee its integrity and stability during a shape shift.

  The elementary building-block of our system is a tendon-driven piecewise foldable chain, composed of many concatenated robotic particles. Given the large number of elements the derivation of static and dynamic models is non-trivial. With particular focus on the dynamics of the system, existing modeling methodologies specific for multibody systems (i.e., chain or tree-type systems [103]), typically address the dynamic problem considering only rigid bodies without compliant elements. Instead, as our chain is a multibody system which contains compliant elements (i.e., the tendons that actuate the chain), the existing methodologies do not fully support our case. Conversely, existing modeling methodologies that address the dynamic problem with complaint elements, such as tendons, typically consider tensile forces being homogeneously applied to a continuous flexible body. Thus, in order to derive a dynamic model of our system, we combine these two methodologies in order to obtain a sufficiently accurate model able to predict the behavior of the system being actuated.

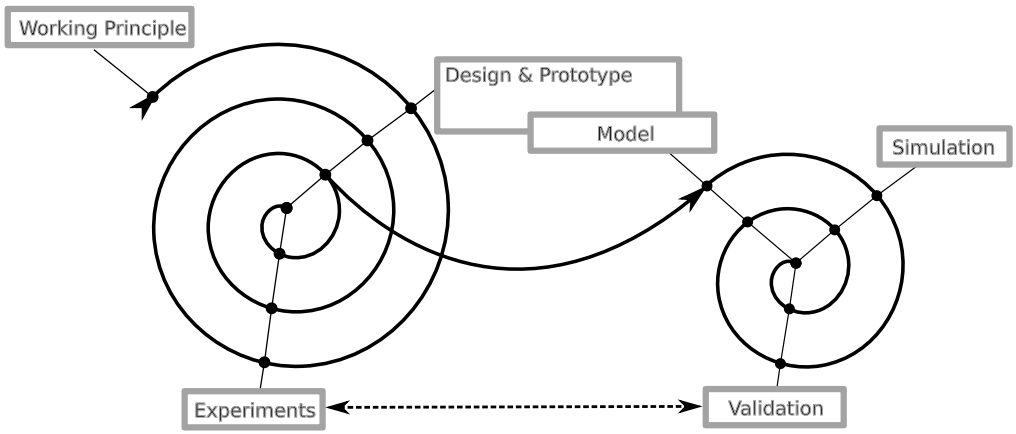


Figure 1.1: Methodology: experimental approach to gradually improve working principle, design and modeling.

1.5 Methodology

The design of a Shape-Shifting Material (SSM), where many robotic particles cooperate to form mutable shapes, raises cross-domain challenges and questions, as discussed in the previous section 1.2. Aiming at studying and solving the major challenges behind the realization of a high-resolution rendering SSM, we devise a novel design principle ("Force-Guiding Principle", see Sec. 3.2.3) and verify its feasibility. We follow a systematic experimental approach based on a cyclical alternating of design, prototyping and experimental phases. The goal is to gradually refine the initial design principle through experiments which demonstrate its actual feasibility. The left spiral in Fig. 1.1 summarizes this methodology.

Once an acceptable prototype is obtained, the next step consists in deriving models to predict the behavior of the system being actuated. These models are required to support planning and control algorithms, which coordinate the actuation of the system during a shape shift. To obtain sufficiently accurate models, we follow a systematic approach to progressively refine the models through three phases: model formulation, simulation, validation. In order to validate the consistency of the models with reality, we compare the experimental results previously obtained to improve the design (i.e., left spiral) with the simulated results that the models produce. The right spiral in Fig. 1.1 summarizes this methodology.

1.5.1 Case-Study: Shape-Shifting Display

In order to support our quest and obtain meaningful insights, we define a case-study that addresses all the major challenges listed above and that helps us verify the effectiveness of our solutions. Our goal is to find solutions that, despite being tailored to the specific case-study, can also be extended and applied to more general cases. To this end, we select as a case-study the "Shape-Shifting Display" introduced in Sec. 1.1, which has the following properties:

- It is a sufficiently generic and representative application of an SSM.

- It presents a well-defined set of requirements, which raise well-defined challenges.
- It has practical implications to uplift interest and motivation also in heterogeneous research communities.

A Shape-Shifting Display is a mechanical device able to approximate the external surface of 3D objects, which can be used, for instance, to "materialize" 3D Computer-Aided Design (CAD) models. It is essentially a shape-shifting surface. Our goal is to obtain high-resolution rendering of reasonably sized shapes. Major challenges are system scalability and particle miniaturization for which novel working principles and software solutions need to be explored.

1.6 Structure of the Thesis

The following chapters are organized as follows.
The second chapter introduces background and existing related work concerning the realization of a Shape-Shifting Material (SSM). In particular, an overview of the potential approaches for the realization of SSMs is followed by a discussion of the existing state-of-the-art architectures specifically designed for SSMs.

Our first contribution is presented in the third chapter, where in order to overcome the limitation of the existing state-of-the-art architectures, we propose a novel design principle and demonstrate its feasibility through experimental validation. Also, to assess the scalability of a system based on our design principle, we derive an upper-bound limit of the possible number of particles that our design principle allows.

Our second contribution is presented in the fourth chapter, where we explain how optimal planning and control algorithms can built upon the combination of static and dynamic models of the system. In particular, we derive static and dynamic models to predict the behavior of our system when actuated. To this end, we need to overcome some limitation of existing modeling methodologies which do not fully support our specific case, where rigid and compliant elements are combined. As the mechanical stability of the system also depends on the derived models, the latter need to be accurate and consistent with reality. Therefore, experimental validation is provided for each model.

As conclusion of this doctoral thesis, the fifth chapter summarizes our results and contributions beyond the state of the art as well as indicates limitations of the proposed solutions. Also, an outlook on future work is provided.

Chapter 2

Background and Related Work

The term Programmable Matter (PM) defines a class of innovative materials able to change their physical properties on demand, in a controlled fashion. The main challenge is to take control over the physical properties, for example, to modify stiffness, transparency, color, and shape. We focus on SSMs, where a shape shift can be achieved by "programming" the interactions among the elementary particles forming the material. Thereby, a shape-shift results from the synergy of many robotic particles that together form a morphable object.

In this chapter, we focus on hardware architectures and mention some software approaches towards the realization of an SSM.

2.1 Shape-Shifting Materials: Approaches

Programmable Matter (PM) represents a class of innovative "software controllable materials", whose shape and physical properties can be changed on demand in a software controllable fashion. The ultimate vision is to obtain a programmable structure composed of computationally limited *elementary particles* able to self-organize and cooperate to achieve a collective goal. An example of existing very basic PM is the flat surface of an LCD monitor that can display images by controlling the color of the units (pixels) it is composed of. In a broader perspective, PM allows the creation of objects whose visual aspects (e.g., color) and tangible features (e.g., shape) adapt to unpredictable situations, either autonomously (e.g., self-repairing materials) or under explicit control of the user.

The realization of PM poses many interdisciplinary questions, concerning in the first place the design of the elementary particles composing the physical infrastructure as well as the strategies to coordinate the potentially large ensemble such particles form. The fundamental challenges that derive from the combination of these two aspects call for dedicated investigation across multiple research domains. Researchers in the fields of biology, chemistry, computer science and robotics, have addressed both the "hardware" and the "software" problems.

Potential solutions based on Bio-Chemical and Modular Robotics approaches lead to promising results. Bio-chemical systems represent the natural embodiment of PM. Many features of living organisms are desirable for Shape-Shifting Materials (SSMs) as well. For instance, the ability of the former to self-heal maps to the ability of the latter to self-repair. The possibility of manipulating organic tissue (e.g., DNA) to produce nano-

metric structures, allows the realization of self-assembling systems. However, the difficulty of synthesizing organic tissue with controllable characteristics makes this approach yet difficult to be exploited on large scale, for general applications. Conversely, established techniques for controlling the modular robots of a modular robotic system are largely available. As modular robots can in principle be scaled in size to become "macro-cells" of artificial tissues, modular robotics is potentially a valuable solution towards the realization of SSM. Even though this approach is also not devoid of remarkable challenges, different architectures have been proposed during the past two decades to create the "robotic particles" forming the artificial SSM. In this perspective, biological and chemical systems offer inspirational paradigms to define the interactions among the robotic particles.

In the following, we discuss some bio-chemical approaches, in order to better understand their potentials and limitations, and also underline the possibility to create hybrid solutions that combine robots and biological organisms. Our attention moves then to modular robotic systems adopted for the realization of SSM. In particular, two specific problems need to be addressed: how to design the modular robotic particles to support the realization of SSM and how to coordinate the collective behavior of such particles to control their overall morphology.

2.1.1 Bio-Chemical Systems

The analogies between the characteristics of biological systems and those we expect for an SSM are quite remarkable. Biological systems are capable of response to stimuli, growth, development, self-healing. Likewise, an SSM should be able to react to external stimuli (e.g., user's input), to grow, to develop new functionality (i.e., shape shift), and to self-repair. In the biological domain, organisms are subject to "biological" rules that define cooperation and interaction of cells necessary to survive and evolve. Similarly, the elementary "particles" composing an SSM obey to programmed rules that define cooperation and interaction models in order to maintain the system integral and control its shape.

The main differences between the two classes of systems are the flexibility to mutate *morphology* and the reaction times. The biological rules that control the growth and other features of an organism are difficult to change as millions of years of evolution demonstrate. Even when an organism is programmed to evolve into a specific morphology, arbitrary long times are required. On the contrary, the morphology of an SSM is expected to be easy to control and the reaction time to be in the order of seconds. Nonetheless, considering the remarkable analogies between the two domains, research in the field of SSM explores solutions based on or, at least, inspired by bio-chemical systems.

Hybrid shape-shifting architectures derive from the combination of biological and robotic systems. The results are innovative architectures capable of self-maintenance with a minimal environmental impact. In their work, Hamann at al. [47] investigate the potential of natural plants to support the creation of bio-architectural structures. Their approach demonstrates how growth can be controlled to obtain desired shapes by means of external stimuli that "override" the naturally programmed development of plants. This has the dual advantage of minimizing the initial investment of resources and also of generating environmental friendly solutions. The main limitations concern the impossibility of achieving fine control, first, over the resulting structures and also on its potential reconfiguration. In addition, the prolonged time required to generate a desired shape and the effort to

preserve such a shape makes this approach unsuitable for general applications.

Bio-inspired approaches attempt to imitate the evolutionary mechanisms that regulate the development of living organisms, which determines their ability to self-heal. To this end, synthetic biology explores how DNA structures can be engineered to self-assemble when suspended in fluids [51]. This allows the creation of complex geometries where DNA forms 2D patterns and 3D wireframe structures [11]. Self-assembly using "DNA motors" [9], establishes a link between bio-chemical systems and cellular-automata [20, 116] which allows the definition of rules regulating the assembly patterns [117]. In this regard, Nagpal et al. [76,77] demonstrate the similarities between multicellular and multiagent systems, where distributed programming methodologies allow complex geometries to emerge.

Although solutions based on DNA have the advantage that the elementary units are nanometric passive elements, the likelihood for the target shape to eventually emerge depends on a stochastic process that makes shape formation difficult to predict. Also, considering that in general a controlled and dedicated environment is required, the solutions that DNA manipulation offers, are not suitable for general applications. These techniques developed to manipulate nano structures, like Deoxyribonucleic Acid (DNA) strands, can nevertheless be exploited for the fabrication of more sophisticated modular robotic particles composing an SSM.

2.1.2 Modular Robotic Systems

The concept of "Self-Reproducing Machine" able to behave like living organisms has been around for more than six decades [83]. Modular robotics [39, 41, 91] has the potential to make this possible, as it combines the strength of engineered robotic systems with the adaptability of natural biological systems. The key idea is that a general-purpose robotic system can be built upon the aggregation of modular robots, which can reconfigure their morphology to enable new functionality. This allows us to build systems that adapt to unexpected situations and carry out tasks not even predictable at design time. Compared to conventional approaches, modular robotics reduces maintenance operations to the replacement of a modular part. When autonomously executed by the system itself, this results in a "self-healing" mechanism. For all these reasons modular robotics has become a fundamental building block towards the realization of SSM [22, 119, 126].

SSMs can essentially be seen as an extreme application of modular robotics, where the modular modules become the elementary particles composing the material. As particle spatial arrangement defines the *morphology* of the material, by controlling the interaction among particles and consequently their relative positions different shapes can be obtained.

In the next section, we consider different existing modular robotic architectures in order to highlight their potentials and shortcomings. To complete this section, we introduce a major distinction between stochastic and deterministic approaches based on modular robotic systems.

2.1.3 Stochastic and Deterministic Systems

We distinguish between two main approaches to control the reconfiguration of systems of modular robotic particles: deterministic and stochastic. For higher resolution SSM, smaller robotic particles are required. Thus, to reduce particle size, a possible solution is to replace

active mechanisms with passive ones. So-called stochastic systems [73, 110, 113, 120] apply this principle to the extent that all the actuation mechanisms are removed from particles which result in passive elements only subject to the action of the environment. Shape formation depends on the likelihood that particles moving in a turbulent environment (e.g., a liquid) eventually merge. For example, in [110] passive flat units move randomly on an air table and occasionally bond to each other; when affine units are sufficiently close to magnetically attract particles merge and the process eventually leads to shape formation.

This type of systems is similar to some bio-chemical systems earlier introduced. Despite the potentially simple and inexpensive particle design, the main disadvantage of this approach lies in the fact that a stochastic process governs shape formation. Multiple attempts prolonged for arbitrarily long times are required before producing valid results. In addition, the specific conditions and the required actuation mechanisms in the environment make this approach not suitable for general application. For these reasons, our focus lies on deterministic systems, characterized by deterministic control of the robotic particles.

2.2 Systems of Robotic Particles: Architectures

The architecture of an SSM defines the way particles interact with each other and their freedom of movement. A central question towards the design of Shape-Shifting Material (SSM) is: "To which extent does the underlying architecture affect the range of achievable shapes?". In this work, we aim to demonstrate that arbitrary shapes can be attained also by means of constrained topologies, which only allow relative displacements of adjoining particles. What motivates our choice is the minimalistic mechanical design that results and that facilitates particles miniaturization for higher resolution rendering.

In this section, we first classify the main existing architectures considering particle topology and spatial arrangement. Concerned about scalability and particle miniaturization, an analysis of some of such architectures underlines the major limitations. To summarize our considerations a taxonomy is provided as conclusion of this section.

2.2.1 Classification

To classify the existing state-of-the-art architectures for SSM, we consider three orthogonal design space "dimensions": bond, connectivity, and particle workspace.

1. **Bond** indicates whether particles form a *detachable* or a *non-detachable* topology. While detachable topologies allow particles and groups of particles to temporarily disconnect from the ensemble and to *autonomously* "crawl" towards different locations (i.e., in order to change their *neighbourhood*), non-detachable topologies constrain particles to a predefined neighborhood and only allow relative displacements among adjoining particles[1]. In the following we refer to this concept with the terms "detachable topology" and "non-detachable topology".

[1]We classify as non-detachable also those architectures that allow only *manual* reconfiguration of the robotic particle topology, i.e., which can not *autonomously* modify their topology.

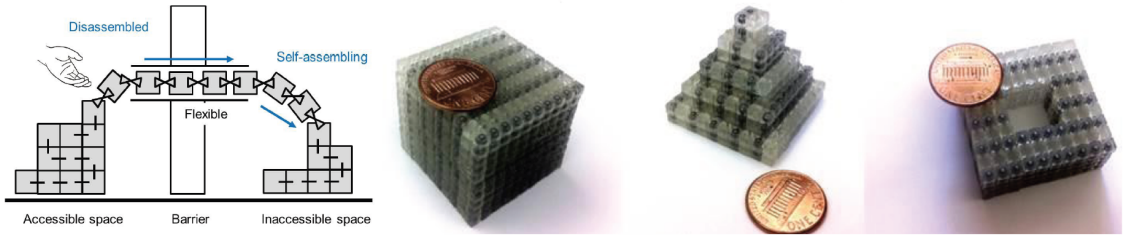


Figure 2.1: SoftCubes [121]: working principle illustrated for the 2D case. The three prototypes show the 3D discrete workspace that particles can reach, despite their linear non-detachable topology.

2. **Connectivity** represents the orientation of the bonds in each particle. In general, literature makes distinction between two cases, which are *lattice* and *chain* type topologies (e.g., as in [82]). However, for a comprehensive and more detailed classification, we consider four cases: *3D-lattice*, *2D-lattice*, *tree-type* and *linear* topologies. Although particles might eventually lie close to each other, we define connectivity on the basis of the bonds among particles and not their proximity after a shape shift (which defines their 'workspace").

3. **Particle workspace** indicates the spatial locations that a particle can eventually reach after a shape shift. The workspace can be *continuous* (any position is possible) or *discrete* (particle position constrained to a grid) and can have three or two dimensions (we omit the 1D workspace that is irrelevant for our scope).

Existing state-of-the-art architectures can be classified based on the defined design space. Table 2.1 groups the principal existing architecture relevant for the realization of SSM and classifies them according to the three design space dimensions. Not all possible combinations of the three coordinates generate meaningful architectures. While for a 3D-lattice topology the workspace is generally also 3D, for other topologies the workspace depends on the Degrees of Freedom (DOFs) of particles. For example, SoftCubes (Fig. 2.1) define a linear non-detachable topology that allows particles to arrange in a 3D-discrete workspace [121], despite the physical constraints.

In the remainder of this section, we analyze some of the architectures listed in Table 2.1. The general assumption is that robotic particles are computational elements, for which power supply and networking are eventually required. As we aim at the creation of a high-resolution rendering SSM, scalability – in terms of number of particles – and minimal particle dimensions are relevant in our analysis. This section ends with a taxonomy that summarizes our observations.

2.2.2 Detachable Topology

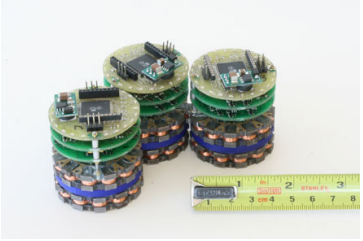
In a detachable topology the robotic modules can freely relocate by means of actuation and latching mechanisms. While this approach allows the formation of literally *any* connected 3D shape, the complexity of the mechanical design complicates particle miniaturization and raises their costs. In addition, particle's mobility makes networking and power supply challenging. This also affects the mechanical robustness of the system, which is related to the strain that the detachable bonds can withstand.

Topology		Architecture	Workspace	Principal
Bond	Connectivity		(system at rest)	Investigator
Detachable	**3D Lattice**	Atron [59, 82]	3D discrete	Jorgensen, Østergaard
		Telecubes [105]	3D discrete	Suh
		M-block [93]	3D discrete	Romanishin
		Miche [40]	3D discrete	Gilpin
	2D Lattice	Macro-Catoms [44, 63, 64]	2D discrete	Goldstein, Kirby
		Micro-Catoms [61]	2D discrete	Karagozler
		Electrostatic-Latch [60]	3D discrete	Karagozler
		Smart-Blocks [74, 88]	2D discrete	Möbes, Piranda
	Tree-Type	M-Tran [75]	3D continuous	Murata
		Polybot [118]	3D continuous	Yim
		Conro [17]	3D continuous	Castano
Non-detachable	**3D Lattice**	Molecubes [125]	3D discrete	Zykov
	2D Lattice	Origami [10, 48]	3D continuous	Benbernou, Hawkes
		Printable Folding Sheet [81]	3D continuous	Onal
		inFORM [36]	3D continuous	Follmer
		Kilobot [94]	2D continuous	Rubenstein
	Tree-Type	Topobo [90]	3D continuous	Raffle
		Posey [109]	3D continuous	Weller
	Linear	Square-tile chain [46]	2D discrete	Griffith
		Cube chain [46]	3D discrete	Griffith
		SoftCubes [121]	3D discrete	Yim
		Millimoteins: [65]	3D discrete	Knain
		Ratchet [111, 112, 115]	3D discrete	White
		SEA Snake [92]	3D continuous	Rollinson
		ChainForm [78]	3D continuous	Nakagaki
		HexRoller [42]	2D continuous	Gilpin

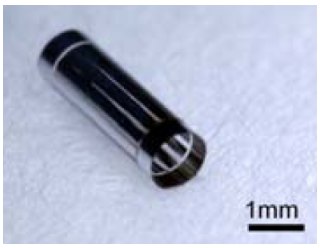
Table 2.1: Classification of some of the existing architectures based on *bonds*, *connectivity*, and *workspace*.

Detachable 3D Lattice Architectures

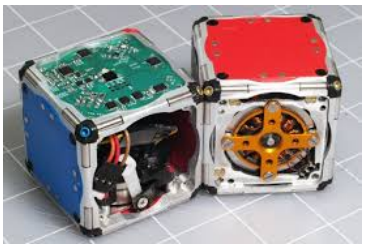
Atron [59, 82] and Telecubes [105] (shown in Fig. 2.2i and in Fig. 2.2h) are a good representative of this class, as they can dynamically disconnect and occupy the vertices of a 3D lattice. Both integrate in a modular block actuators, detachable bonds, controllers. An Atron module consists of two semi-spheres which can rotate about the same axle. Mechanical hooks are regularly arranged around the module to engage up to eighth other modules. The relative rotation of the two semi-spheres causes a spatial rearrangement of the modules connected to each semi-sphere. Similarly, Telecube is a cubic block which can bind to other blocks through magnetic interfaces placed at each face. As the six faces



(a) Macro-Catoms



(b) Micro-Catoms

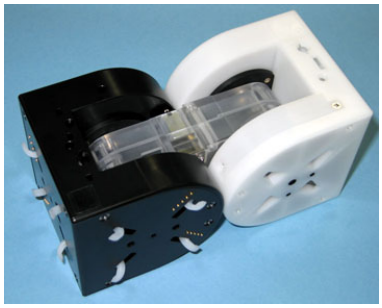


(c) M-Blocks

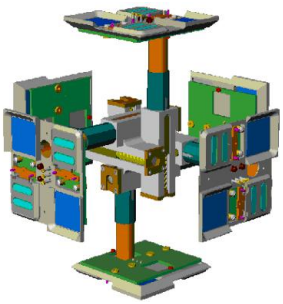
(d) Molecubes

(e) Miche

(f) SEA-Snake

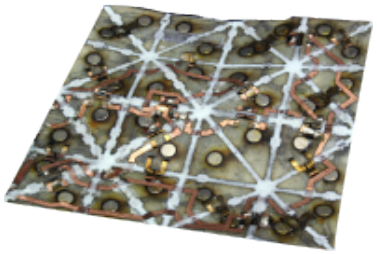


(g) M-Tran

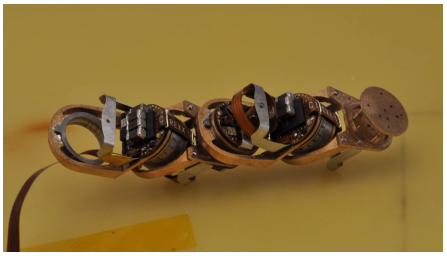


(h) Telecubes

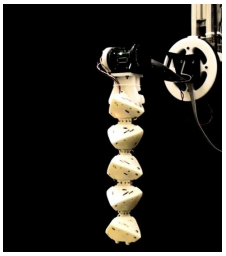
(i) Atron



(j) Origami Sheet



(k) Millimotein



(l) Ratchet14

Figure 2.2: Some of the existing architectures and prototypes.

of a Telecube are telescopic, which means that they can extend towards the six spatial directions, adjoining Telecubes can modify their relative distance. As the combination of magnetic bonds and extensible faces allow Telecubes to dynamically engage other blocks and relocate, an ensemble of Telecubes can form different shape in a 3D discrete lattice. Although both Atron and Telecubes are potentially suitable for the realization of a Shape-Shifting Display (SSD), their mechanical complexity might complicate miniaturization, limit scalability, and raise costs.

To eliminate moving parts, Miche [40] (Fig. 2.2e) exploits magnetic forces to combine actuation and latching. The system consists of cubic particles initially arranged in a solid 3D geometry (e.g., a big rectangular parallelepiped). Permanent magnets on each face of a particle ensure adjacent particles to adhere and maintain their connection. When actuated, a repulsive electromagnetic force overcomes the permanent magnetic fields and split particles apart. In this way, particles are selectively "sculpted" from the initial geometry in order to obtain a target shape. This approach results in a simple mechanical design, as it does not require moving parts. However, a shape reconfiguration is only possible by *manually* restoring the initial solid 3D geometry, which makes this solution unsuitable for a self-reconfigurable system.

Likewise, M-block [93] exploits magnetic forces to retain the final 3D lattice shape that the cubical M-Blocks eventually form. The ability of this system to modify its morphology consists in the fact that M-Blocks can leap to join a different neighborhood. By exploiting the inertial effect internally generated by suddenly breaking a flywheel, these momentum-driven blocks can pivot along an edge or jump to a target location. Despite the simple working principle, challenging algorithms and models are required to control and predict the dynamics of the system. In addition, the internal mechanisms are difficult to reduce in scale for particle miniaturization, and the magnetic bond might limit the scalability of the system, for example, when forming "long-limbed" shapes like a cantilever.

Detachable 2D Lattice Architectures

Also based on magnetic coupling, Catoms (shown in Fig. 2.2a) allow relative movement and latching operations without motors nor moving parts [44, 63, 64]. Cylindrical particles adhere to each other and re-arrange by means of electromagnets regularly arranged around the lateral surface. Similar to a stepper-motor, the sequential activation of the electromagnets generates a rotating magnetic field, which induces adjacent particles to roll about each other. In order to generate sufficiently strong magnetic forces, this solution results in large particles which are difficult to miniaturize.

Aiming at higher miniaturization, Karagozler et al. demonstrate how electrostatic latching solutions [60] can be applied to millimetric-scale tubes (1 mm diameter), in order to obtain "micro Catoms" [61] with a mass of 0.08 mg including electronics (Fig. 2.2b). Electrostatic forces between neighboring Catoms cause the latter to roll about each other and hence relocate. Interesting is the manufacturing process that such tiny straw-like structures undergo, based on a micromachining solution. Power is supplied through two metallic rings, wrapped around the bases of the cylinder. Despite the small size, even suitable for further miniaturization, actuation and adhesion are limited by the relatively weak electrostatic forces. In particular, if we consider a prolonged configuration like a cantilever, the electrostatic forces acting at the thinnest cross-section might not be sufficient to withstand the stress that this specific configuration induces. This limits the scalability of the system.

Towards the creation of a reconfigurable conveyors, Möbes and Piranda et al. [74, 88] propose Smart-Blocks. With a cubic shape and a conveying face, Smart-Blocks can modify their relative location to dynamically create paths for carrying microparts. Adjoining modules can reconfigure in a 2D discrete workspace by shifting against each other actuated by electro-permanent magnets embedded in each module. While Smart-Blocks have

high freedom to arrange into flat geometries, this solution requires a planar surface on which modules can slide, which limits the suitability of the system for the creation of SSD. In particular, when removed from their planar surface to create, for example, vertical geometries, the electromagnetic force holding modules might not be sufficient to ensure mechanical stability to the system.

Detachable Tree-Type Architectures

An M-Tran [75] module consists of two semi-cylindrical boxes linked together, which can pivot against each other and dynamically engage other modules to form different tree-type and linear configurations in a 3D continuous workspace. Each module integrates actuators, detachable bonds, controllers. Multiple M-Tran modules form an electrical grid that supports serial communication and power supply. The relative pivoting of the two halves of an M-Tran module allows adjoining particles to control their angular displacement and thereby to modify the overall geometry of the system. As the workspace of the system is *3D continuous*, locomotion is possible.
M-Tran is in principle suitable for the creation of SSM. However, while the freedom of particles to relocate and thus to form arbitrary 3D shapes is an advantage, the mechanical complexity hinders particle miniaturization towards the creation of high-resolution rendering SSM and raise costs.

2.2.3 Non-Detachable Topology

In a non-detachable topology, the robotic modules are permanently interconnected and only relative (generally angular) displacements of adjoining particles are permitted. While this approach can potentially reduce the mechanical complexity of particles, thereby facilitating their miniaturization and reducing their cost, shape formation is also constrained. Nonetheless, particle's permanent interconnection facilitates networking and power supply as well as potentially increases the mechanical robustness of the system.

Non-Detachable 3D Lattice Architectures

Molecubes (Fig. 2.2d) integrate in a modular cubic block actuators, *manually* detachable bonds, controllers, and communication modules [126]. Externally powered, Molecubes form an electrical grid, where the modular elements supply power to each other. The system can be manually assembled into a 3D lattice topology, although the most convenient configuration is obtained by forming a tree-type topology. The relative rotation of the two halves of a molecube, along the axle coinciding with a long diagonal of the cube, allows adjoining particles to control their angular displacement and thereby to modify the overall geometry of the system. Although the system is designed for manual assembly, Molecubes can in principle support the creation of SSM. However, while the freedom of particles to form arbitrary 3D shapes would facilitate this task, the mechanical complexity hinders particle miniaturization towards the creation of high-resolution rendering SSM.

Non-Detachable 2D Lattice Architectures

Self-folding origami [48, 81] can be built by regularly arranging active robotic particles on the flat surface of a sheet. When actuated, the robotic particles generate regular crease patterns on the surface of the sheet that folds. To retain the target shape, latches (e.g., magnets [48]) are embedded onto the surface of the system. Although the fixed topology facilitates particle networking and power supply, the scalability of the system (i.e., the extension of the sheet) is limited by the maximum forces that embedded actuators and latches can exert. Considering that forces are generally proportional to the size of actuators, larger sheets call for thicker actuators. As multiple folding layers are typically required [10], the maximum thickness of sheet and actuators is constrained. This limits the strength of actuators, and thereby the scalability of the SSM.

"inFORM" [36] is a programmable tangible surface composed of controllable pins, rising from a table. The robotic particles, namely the pins, are regularly arranged in a 2D lattice. A shape-shifting surface is obtained by controlling the relative heights of the pins. The vertical motion of pins can be exploited to actuate objects (e.g. a rolling ball) placed on the surface. With the aid of a projector installed on the top of the surface, the system can be used to display colored 3D objects (e.g., 3D histogram). In addition, a kinetic system installed beside the projector allows users to "draw" 3d objects by hovering their hands above the surface. However, as each pin is individually actuated by a dedicated actuator, the overall costs rise significantly when aiming at the creation of higher-resolution rendering surfaces. Furthermore, the system is hard to scale mainly because of the space required to arrange all the actuators, whose footprint is roughly 1.5 times the size of a pin. Power supply is also an issue for scalability, considering that the actuation of the 900 pins installed in the existing inFORM requires theoretically up to 2700 W.

Nevertheless, a *remarkable insight* comes from this approach: external actuation eliminates the limiting relation between actuation forces and dimensions of the robotic particles. In particular, pins can be easily miniaturized for higher resolution without compromising the functionality of the system.

Non-Detachable Tree-Type Architectures

A chain topology consists of sequentially concatenated particles. A tree-type topology consists of interconnected chains and can thus be considered a generalization of chain topology. Topobo [90] and Posey [109] are example of tree-type topologies. Both designed for educational purposes, they consist of modular links with variable connectivity that are manually assembled, for example, to outline the skeleton of a puppet. The two systems can sense a sequence of poses and memorize it. Goal of Posey is to reproduce such a sequence in an animated application, thus providing a model-and-animate interaction paradigm. Topobo is able to physically reproduce a sequence of movements by controlling the joints of the interconnected elements. As in folding sheets, the local actuation of the joints of Topobo limits the scalability of the system to a maximum extension. Due to leverage effects indeed an increasing effort has to be expected in case of long-limb extensions.

To some extent, also Molecubes (Fig. 2.2d) can be considered a tree-type topology. However, as earlier discussed in Sec. 2.2.2 scalability and particle miniaturization are limited because of actuators being embedded into particles.

Non-Detachable Linear Architectures

In linear topologies (chains), the relative angular alignment of consecutive particles is controlled to fold the chain into arbitrary 2D or 3D shapes. However, as with sheets the length of the chain is limited by the torque that actuators can exert, which means that particle dimensions cannot be arbitrarily reduced and the length of the chain cannot be extended beyond certain limits. Solutions based on external actuation mitigate this problem.

Aiming at the creation of snake-like robots, mainly for rescue and inspection missions, Rollinson et al. propose *SEA Snake* [92] composed of modular 1-DOF elements that can be arbitrarily connected to extend its length and augment its dexterity. The dense integration of electronic components impedes the miniaturization of the modular elements below diameters smaller than 5.1 cm as specified in [92]. This makes this solution not suitable for our goal.

Similarly, ChainForm [78] exploits the ability of a chain to outline arbitrary shapes to create a reconfigurable display, where arrays of LEDs can be controlled to show, for example, a character. The flexibility of the chain makes the display attachable to a large variety of surfaces and shapes. The integration of servomotors in the system raises costs and limits the miniaturization of the modular elements and the scalability of the chain.

Millimoteins (Fig. 2.2k) are motorized chains that can outline 3D curves in a discrete workspace [65]. According to its designers, the dimension of a single particle can scale down to $1\,mm$ edge. SoftCubes [121] can also fill a 3D-discrete workspace using magnets embedded in particles. The generation of a 3D-discrete workspace using these types of chains is formally proven in [46]. Limited to a 2D-discrete workspace, Griffith et al. [46] also demonstrate the potential of a 2D square-tile chain to form arbitrary flat shapes. As all the above approaches require embedded actuators acting at the hinges of the chain, the scalability in terms of maximum number of particles is constrained by the limited maximum force such actuators can exert.

External Actuation. An important distinction needs to be made between locally and externally actuated particle chains. Externally actuated chains enable high scalability – in terms of number of particles – as the external actuator can be upgraded to support a growing number of particles. At the same time, this approach facilitates particle miniaturization. For instance, as earlier observed, "inForm" demonstrates the latter concept.

White et al. [111, 112, 115] use an external manipulator to fold a chain (Ratchet14 in Fig. 2.2l) in a 3D-discrete workspace. A remarkable advantage is that particles in the chain present a minimal design and therefore are amenable for miniaturization. In addition, as the strength of the external manipulator can be increased without modifying the chain nor the particles, an arbitrarily large number of particles can be actuated. Consequently, external actuation facilitate system scalability. The possibility to improve both chain scalability and particle miniaturization by means of an external actuator allows high-resolution rendering and contains costs. However, aiming at the realization of a Shape-Shifting Display (SSD), the solution that White et al. propose is not suitable "as it stands" for our scope because it requires a dedicated actuator per chain. As in our solution, later presented in Chap. 3, multiple foldable chains are combined to form a shape-shifting surface (i.e., the SSD), the requirement of a dedicated actuator per chain limits the scalability of the whole system (Chap. 3 clarifies this concept).

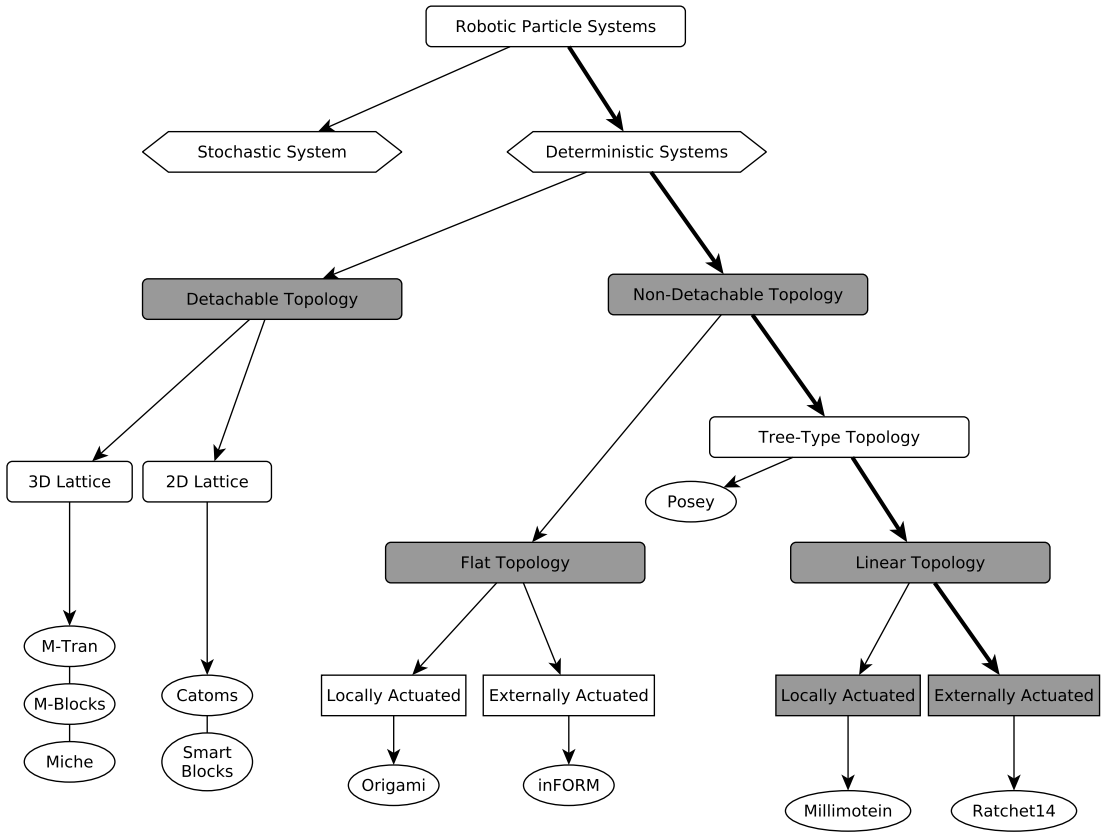


Figure 2.3: Architecture Taxonomy for Systems of Robotic Particles.

2.2.4 Taxonomy

The taxonomy shown in Fig. 2.3 summarizes the considerations derived in this section. The gray blocks will be reconsidered in Chapter 3 to motivate our design approach.

2.3 Shape-Shifting Algorithms

In order to successfully perform a shape shift, planning algorithms are required to coordinate the ensemble of particles and to handle the inherent complexity of a shape-shift operation.

Only a limited amount of work on algorithms targets programmable matter [34]. While the main challenge of such algorithms lies in the coordination of several robotic particles with limited computational capabilities, they also need to deal with networking, distributed control, optimization, faulty elements, particle mobility [30, 79]. Focussing on the particle coordination problem, programming approaches range from abstract concepts like *cellular automata* to define the local rules that each particle executes, to more global and high-level primitives that define the target shape, for example, in terms of expected growing directions.

In a lattice detachable topology, shape shift is performed by migrating single parti-

cles or groups of particles. Because of particle mobility, the system state space is quite large. To efficiently handle this complexity, distributed algorithms are generally adopted to coordinate the collective actuation. Algorithms control the movement and the internal configuration of the particle, and need to guarantee the integrity of the system by satisfying the physical constraints imposed by the hardware implementation. In addition, they have to deal with decentralized, asynchronous and parallel computation [79].

Conversely, in a non-detachable topology, shape shift is performed by controlling the relative displacement of adjoining particles. While on the one hand this reduces the state space of the system, on the other hand shape-shifting algorithms have to deal with the geometrical constraints and yet guarantee system integrity and stability. In Chap. 4, we elaborate on this aspect considering that, when not properly planned, a shape shift might cause too intense actuation forces to destabilize the system.

Hou et al. [53] demonstrate that the optimal reconfiguration problem for detachable tree-type architectures, such as M-Tran [75], is in class of complexity *NP-complete*. They address the optimal planning problem by means of a graph [45, 52] whose vertices correspond to the possible states of the system. Paths on the graph map to optimal planning solutions in exponential time and also to sub-optimal solutions in polynomial time. However, as the proposed algorithm does not take into account some physical constraints (e.g., maximum forces that the system can exert to reach a target configuration) and other hardware limitations, complementary research and experiments are needed to validate this approach.
Concerning detachable 3D and 2D lattice architectures and non-detachable architectures, it is not yet possible, to the best of our knowledge, to assert whether the class of complexity of the optimal reconfiguration problem is also NP-complete.

In the remainder of this section we reference some relevant shape shifting approaches found in literature, with the aim of providing a general overview of available solutions. As a strict association between shape shifting approaches and Shape-Shifting Material (SSM) architectures is not always possible, we group such approaches into four categories distinguished by a common objective. We notice that, roughly, the first two categories are mainly suitable for (but not limited to) detachable topologies, while the last two categories are mainly suitable for (but not limited to) non-detachable topologies.

i. **Group Coordination.** Existing approaches are mainly suitable for detachable topologies and address the coordination problem by splitting the set of particles into subsets called "meta-modules" [16, 22, 97, 107] or more generally by adopting the concept of hierarchical motion [12]. Other solutions focus on "hole motion" [25] rather than directly controlling the particles. High-level global primitives such as "gradient" and "pheromone" [22] or "growing points" [23, 104] indicate the growth direction of the system of particles.

Similarly, Piranda et al. [87] propose a fully distributed shape-shifting algorithm based on motion rules and meta-modules suitable for 2D lattice-based systems. In particular, the solution elaborates the sequence of intermediate configurations necessary to obtain a target arrangement of square tiles that satisfies the physical constraints. The latter impose that each particle must have at least an edge in common with another particle in the system. The proposed algorithms can be adapted to different types of hardware implementation. In particular, it is suitable for the Smart-Blocks described in Sec. 2.2.2.

ii. **Logic Predicates.** A target shape can be described by means of logic predicates that define the expected final state of the system. For example, aiming at the creation of an arbitrary loop of particles, a predicate that describe this condition is: "each particle in the system must eventually connect to two and only two distinct particles". Particles in the system asses the predicate in a distributed fashion, and thereby modify their *local state* to satisfy the predicate (e.g., being connected to other two particles). Eventually the state of the whole system satisfy the predicate.

Meld [4,5] and LDP [26,27] support this approach. Meld is an ensemble-oriented logic programming language that supports the automatic generation (compilation) of distributed code. It is based on a set of logic rules and facts that represent the status of the system. The application of rules generates new facts, and the combination of matching rules and facts generates actions which eventually actuate particles.
LDP (Locally Distributed Predicates) is a declarative programming language base on the concept of "pattern matching". Predicates are essentially based on the status of the nodes and on their topology. The actuation of the system is defined through actions, triggered by matching predicates. Originally designed to detect error conditions in large multi-robot systems [24], LDP supports distributed debugging.

iii. **Cellular automata.** Hybrid cellular automaton approaches [29, 38, 51, 71] allow the creation of arbitrary complex patterns by encoding the assembly rule in the edge of tiles [116] and by controlling the condition of the environment in which the abstract tiles exists [29]. This causes tiles to join each other according to the encoded rules (e.g., edge affinity), whereby a desired pattern eventually emerges. This method, that solves the generic problem of paving (coating) a certain area, can be generally applied to self-assembly systems. Assuming that the physical constraints of real hardware topologies could be encoded in tile-matching rules, the approach can be exploited to define, for example, the folding sequence (i.e., angles at the hinges) of concatenated particles.

iv. **Geometric approaches.**

Cheah et al. [19] propose a region-based shape control to coordinate a group of particles forming a detachable 2D lattice topology that arrange in a target 2D geometry. The optimization problem is split into a global and a local objective: the former defines the target region where robots should eventually lie; the latter defines the minimum distance among robots. As robots competitively try to achieve both such objectives, the target configuration eventually emerges.

Concerning self-folding origami, Huzita et al. [54] have presented in 1998 six axioms which represent the basic folding rules of origami sheets. In 2001, Koshiro Hatori has added the seventh axiom that completes the set. These axioms essentially define the folding primitives based on simple geometric entities, such as lines and points, to describe any folding shape that origami sheets can achieve. Through these axioms, Lang [67] could demonstrate that the folding patterns to approximate arbitrary 3D shapes can be derived from the 2D projection of the skeleton of the target object. Such a projection indicates the principal folding points and patterns to eventually obtain the target shape.
Independently from other approach, Benbernou et al. [10] propose a universal folding algorithm suitable for the creation of target shapes that can be decomposed into cubes.
As an origami sheet consists of programmable particles regularly arranged on its surface,

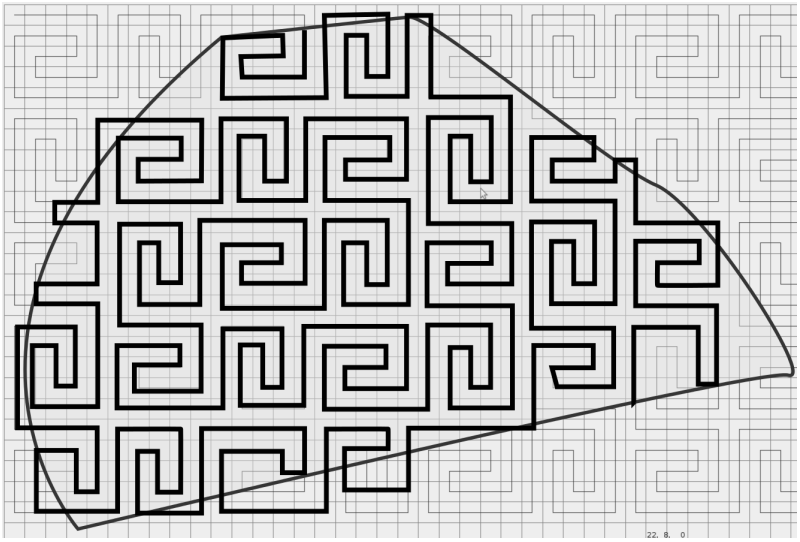


Figure 2.4: Example of a space-filling curve approximating a 2D region. The curve extends beyond the boundary of the target region and needs to be "cut" and "reconnected".

to support shape formation and distributed code generation, Origami Shape Language [76] describes by means of distributed rules the folding crease patterns. Such rules are based on the folding axioms [54].

Griffith et al. [46] exploit the concept of 2D and 3D space-filling curves to identify a possible folding path for linear architectures, in order to approximate the area or the volume of a target shape. The space-filling curve is defined in order to match the physical constraints that the system imposes. This method applies to *flat* square-tile particle chains and to cubic particle chains [21], such as SoftCubes in Fig. 2.1. An example of space-filling curve that approximates a region is given in Fig. 2.4.

2.3.1 Discussion

As described in Chap. 3, our design defines a shape-shifting surface composed of multiple chains, where each chain approximates the contour of flat geometries in a 2D discrete workspace. In order to plan a shape shift, a "geometric approach" is required to indicate the target folding state of consecutive particles in each chain.

Considering the geometric approaches presented above, namely region-based and origami, they are both not suitable for non-detachable linear topologies. The first approach targets 2D detachable topologies and requires particles to move in a 2D continuous workspace. The second approach, origami, targets folding sheet that can support multilayer folding. In our case, the shape-shifting surface that we obtain from the combination of multiple chains is not suitable for multilayer folding. In addition, the application of the folding axioms requires the folding sheet to be free on the four sides, while our surface is instead physically constrained on one side.

The third geometric approach based on space-filling curves is suitable for linear topologies to infer a folding path that fills the inner region of flat geometries. However, it becomes

computationally inefficient to approximate only the contour of a flat geometry. As Griffith et al. [46] describe, a 2D space-filling indicates the path of a chain covering a 2D connected region. A generic space-filling curve [98] presents fractal properties that make the curve expand over an infinite large space. For example, the space-filling shown in Fig. 2.4 extends beyond the borders of the target 2D region. In order to only consider the portion of curve that lies within the target region, Griffith's approach consists in "cutting" and reconnecting the space-filling curve along the border of the target region. This approach is justified in Griffith's scenario as it provides the whole folding path at the cost of the cutting operations along the border. Instead, it becomes computationally inefficient when targeting only the border of a flat region, for which the "cut" and "reconnect" operations are dominant.

2.4 Enabling Technologies

Many enabling factors and technologies have paved the way towards the realization of programmable matter. With particular focus on modular robotic particles, we consider the main technological results and directions that let us anticipate that large-scale systems of micro and nano-metric robotic particles will be feasible in the future.

2.4.1 Large Systems of Nanometric Robotic Particles

The technological advancements that have led to the continuous increase of processing power and the realization of Micro-Electro-Mechanical Systems (MEMS) and Nano-Electro-Mechanical Systems (NEMS) support the transition from macro-modules to micro-particles [13] necessary for high-resolution Shape-Shifting Material (SSM).

Chemically Assembled Electronic Nanotechnology (CAEN), together with Nanotechnologies and Bio-Engineering, allows us to manipulate materials and biological tissue at the nano-scale level enabling the creation of extremely tiny structure suitable for the realization of self-assembled components. Although the latter typically present limited capabilities, more sophisticated modular robotic particles can be built through a self-assembling process.

2.4.2 Dense Networks of Computational Elements

Research on Wireless Sensor Networks (WSNs), pervasive computing, and Internet of Things (IoT) has addressed the problem of how distributed intelligence can emerge from dense networks of small low-power embedded devices. Although this has mainly been studied on the large scale, the results can be adapted to smaller scales of densely networked modular devices, such as the robotic particles forming an SSM.

Swarm robotics [95, 99] demonstrates how the coexistence of computationally-limited agents can be exploited to decompose a complex global problem into simpler local problems that agents can autonomously tackle and solve. In a similar way, the cooperation among the robotic units of an SSM leads to the global solution (shape shift) to emerge.

Closely related to programmable matter, Amorphous computing [2, 6] defines the concept of spatial programming which faces the problem of coordinating an ensemble of irregularly placed, asynchronous, interacting, computing elements. The fundamental concept

is that the collection of computing elements can be exploited as a "medium" to propagate the programming rules to neighboring elements. To some extent, this concept extends the cellular automaton approach to real physical system [6].

2.4.3 Shape-Memory Alloys

The use of electroactive polymers to replace conventional actuators in modular robotic particles, contributes to reducing the volume of the modules for the absence of moving parts, like electric motors. In particular, Shape-Memory Alloys (SMAs) [84] represent a valid alternative to other types of actuators when contained dimensions are relevant. SMAs have the ability to recover an initially memorized geometry when heated above its transformation temperature. In its simplest application, SMAs can be used in form of wires that shrink when heated, for example, by means of an electric current. Considering, for instance, that a 100 μm diameter wire can exert a nominal contraction force of about 15 N, SMAs allows the creation of compact actuators. Compared to conventional actuators, the compactness and the energy density (i.e., power-to-weight ratio) of SMA actuators outperform by at least one order of magnitude conventional DC and AC electric motors [50], making the SMAs an enabling technology with a wide spectrum of potential applications. The use of an SMA to locally actuate modular robotic particles of an SSM, minimizes the volume of the robotic elements and simplifies the mechanical design considering that the SMA can be easily integrated into miniaturized modules [122]. However, a major drawback is the limited displacement that an SMA actuator (e.g., wire) can produce of about 4% of its length. This means that to produce a significant movement of the actuated parts, leverage mechanisms are necessary to amplify the actuation effect.

A profitable use of SMAs is in combination with auxiliary actuation mechanisms. In order to exploit the reduced displacement that can be obtained with an SMA, we devise a system (described in Chap. 3) based on a tendon-driven solution to remotely actuate each particle by means of an external actuator. This idea is grounded in the working principle presented by White et al. [111] and extends it.

2.4.4 Tendon-Driven Robotics

Tendon-Driven Robotics is widely adopted to realize lightweight and powerful robotic structures (e.g., robotic arms). In conventional articulated robots, actuators are directly embedded in the joints interconnecting the rigid element of its skeleton, to actuate and drive the position of the end-effector. In order to obtain thin and lightweight articulated structures, bulky actuators at the joints are replaced by passive mechanisms remotely actuated by means of a set of tendons deployed inside cavities in the articulated structure.

In this way, as the active actuators can be placed outside the robotic arm, the size, the weight and the mechanical power of the resulting system can be optimized. Removing actuators from the arm has the dual effect of lightening the structure of the robot for better dynamic response, and simplifying the maintenance of the actuators, generally critical for a robotic system.

Chapter 3

Tendon-Driven Force-Guiding Particle Chain

This chapter describes our design approach towards the realization of a high resolution SSM using modular robotic particles. Even though the ideal scenario depicts a universal SSM suitable for any application, the design of an SSM is very challenging also when targeting a single specific application, because of the conflicting requirements that arise. To help us elaborate on this concept, we define as target application the creation of a high resolution Shape-Shifting Display (SSD). This is essentially a mechatronic device that approximates the surface of arbitrary 3D objects and provides a tangible representation of the object. The application is sufficiently generic to encompass the major challenges common to other SSM applications, such as *formation of arbitrary shapes*, *scalability* and *particle miniaturization*.

As the previous chapter highlights, a conflict exists between system scalability (i.e., the number of robotic particles) and particle miniaturization. In order to obtain high-resolution rendering, a large number of tiny particles is required. This calls for strong actuators able to withstand a growing number of particles, whose size diminishes in favor of higher resolution. However, as size and strength of actuators are generally correlated, the integration of strong actuators in tiny particles is particularly difficult.

We overcome this conflictual situation by introducing the "Force-Guiding Principle". Instead of generating the actuation forces by means of actuators embedded in particles, we exploit an externally provided force to actuate each particle. A built-in mechanism (weak and amenable for miniaturization) guides the external force to actuate a particle. In this way, system scalability and particle miniaturization become possible.

This chapter begins with an analysis of the main requirements of an SSD. This leads to a breakdown of the major conflicts that exist, for which solutions are devised in the rest of the chapter.

3.1 Requirements

In this section we analyze the main requirements of a Shape-Shifting Material (SSM). To facilitate this task, we target the realization of a Shape-Shifting Display (SSD) as a concrete case-study. This is a representative application of SSM, which encompasses the

major challenges common to many other applications.

Our goal is to highlight the major conflicts that the choice of a specific architecture causes at different abstraction levels (software and hardware) and also to demonstrate that, by constraining particle freedom of movement, it is possible to improve system scalability, facilitate particle miniaturization, and nonetheless allow the formation of arbitrary shapes.

3.1.1 General Overview

Considering the many potential applications introduced in Ch. 1, the development of a general purpose SSM is a challenging task. Even when focusing on a specific application, many conflicting requirements arise at different abstraction levels. To aid our elaboration on that, we consider the realization of a Shape-Shifting Display (see Sec.1.1) and outline a well-defined set of challenges. Through a preliminary top-down analysis, we identify the main requirements and determine the *vertical* dependencies that link requirements at distinct abstraction levels. With a more accurate analysis, we highlight the horizontal dependencies (i.e., among the requirements at the same abstraction level) which cause critical conflicts. In particular, we argue that such conflicts mainly depend on the selection of the underlying architecture. Our ultimate goal is to devise solutions to minimize the negative effect of these conflicts.

A Shape-Shifting Display (see Sec.1.1) is a device able to take on arbitrary 3D shapes, for example, in order to "materialize" digital CAD models.

Defintion: A Shape-Shifting Display (SSD) is a mechatronic screen that approximates the surface of arbitrary 3D objects. It enables *high-resolution rendering* of *connected 3D surfaces*, by spatially re-arranging the modular robotic particles composing its structure. In order to reproduce 3D elements at a *reasonable scale* (i.e., overall size), the physical implementation of such robotic system should ensure *scalability* as well as *integrity and stability*.

The above definition indicates the high level requirements of an SSD. In analogy to an existing device, such as a computer monitor, the high level requirements (e.g., resolution) imply lower level requirements (e.g, pixel size) that eventually drive the physical design and implementation. With a similar approach, we derive lower level requirements from the high level ones and divide all the requirements among three distinct abstraction levels, which we call: (I.) *application requirements* (high level), (II.) *system design requirements* (intermediate level) and (III.) *implementation requirement* (low level).

Fig. 3.1 shows the three abstraction levels and the corresponding requirements. The *application* level includes the "macro" properties of an SSD, which map into requirements at the *system* level, in turn calling for requirements at the *implementation* level. We proceed with an analysis of the requirements at the application and system levels and derive their mutual dependencies.

Application and System Design Levels. Each block in Fig. 3.1 indicates a specific requirement and the adjacency between piled blocks indicates *interdependency*. For example,"High-Resolution Rendering" (application requirement) calls for "Particle Miniaturization" (system design requirement). For each of the four application requirements depicted in Fig. 3.1, the implications on system design requirements are summarized as follows.

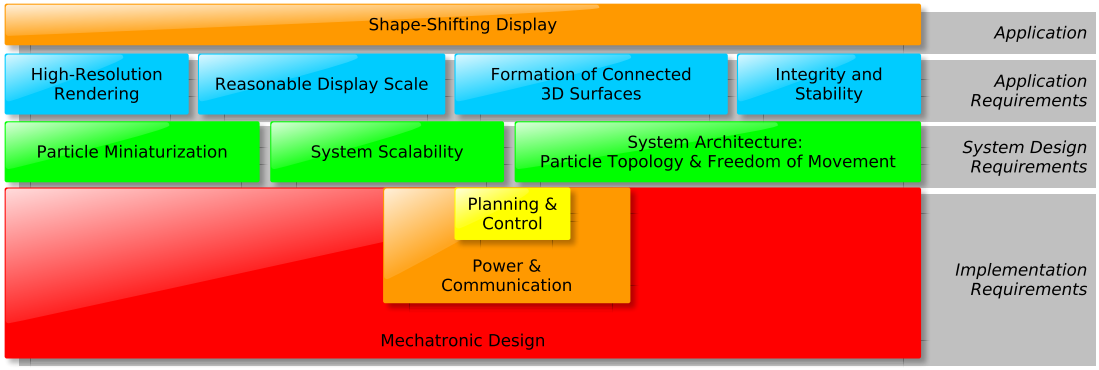


Figure 3.1: Requirement stratification: at different abstraction layers requirements raise conflicting challenges.

1. **High-Resolution Rendering** indicates the ability of an SSD to reproduce high-fidelity objects. This depends on the number of particles per unit of volume, which we call *ppv* – i.e., particles (voxels) per unit of volume (cm^3). As the resolution is inversely proportional to the particle dimensions, this implies the requirement "Particle Miniaturization" at the system design level.

   $\hookrightarrow$ **High-Resolution Rendering** $\implies$ **Particle Miniaturization**

2. **Reasonable Display Scale**. Similar to a computer monitor, to render reasonably scaled objects the display area should be relatively big, an SSD should ensure a minimum display volume. We indicate this property considering the volume of the SSD workspace, namely the overall volume that the particles composing the SSD can reach.

   Considering the minimal particle dimensions that derive from Spec. 1, a large number of particles need to be combined to obtain a reasonably sized SSD. This calls for high scalability (many particles) and mutually depends on particle miniaturization.

   $$\hookrightarrow \textbf{Reasonable Display Scale} \begin{cases} \implies \textbf{Sytem Scalability} \\ \iff \textbf{Particle Miniaturization} \end{cases}$$

3. **Formation of 3D Connected Surfaces.** The ultimate goal is to build an SSD that approximates the surface of arbitrary 3D objects. A surface can be geometrically classified considering four classes having increasing complexity: *connected convex*, *connected concave*, *connected perforated*, *disconnected*. In general, it is possible to approximate a high complexity class by means of a lower complexity class. For example, *disconnected* surfaces can be approximated to a *connected perforated* surface, where the gap separating the two disconnected parts is a large hole extending beyond the borders of the perforated surface. Similarly, a *connected-perforated* surface can be approximated to a *connected concave* surface, where hollow cavities approximate through-holes. This limits the classes of geometries to be considered to connected 3D surfaces. This means that an SSD able to display 3D *connected convex* surfaces can display, or at least approximate, the surface of arbitrary 3D objects.

Shape formation affects primarily the choice of the underlying architecture that defines the interaction among the robotic particles, their topology and their degrees of freedom to spatially arrange into shapes. Sec. 2.2 elaborates on this concept by taking into account the ability of the four principal state-of-the-art architectures to form shapes and also the implication that the choice of an architecture has on other aspects of the system.

As the formation of arbitrary shapes relies on a minimum number of particles to produce a shape, the high scalability of the system is also a requirement.

$$\hookrightarrow \textbf{Formation of 3D Connected Surfaces} \implies \begin{cases} \textbf{Sytem Architecture} \\ \textbf{System Scalability} \end{cases}$$

4. **Integrity and Stability**. Mechanical robustness is a critical aspect for SSMs, especially in safety-critical applications like *reconfigurable environments* (Sec. 1.1). Integrity is the ability of an ensemble of particles to withstand external forces in order to remain connected. Stability is the ability of an SSM to properly operate and to retain a target shape in spite of the external forces. As in both cases the number of particles is relevant, we measure integrity and stability as the *ratio* between the maximum external force F_{max} the system can withstand and the number N of particles: $\frac{|F_{max}|}{N}$. As, typically, the forces necessary to permanently compromise the integrity of the system are higher than the forces that temporarily destabilize the system, we define $\frac{|F_{max}|}{N}$ as the *stability index* that indicates the overall mechanical robustness of an SSD: the higher this index, the higher the stability.

   The selection of a system architecture affect the robustness of the system. For example, detachable topologies that allow particle relocation, present more challenges concerning robustness compared to non-detachable topologies that only allow relative displacements of permanently joint particles.

   $$\hookrightarrow \textbf{Integrity and Stability} \iff \textbf{Sytem Architecture}$$

Hardware and Software Components. Fig. 3.1 depicts the *implementation requirements* at the lowest abstraction level. At this level hardware and software define the physical implementation of the system and in particular the structure of a particle as well as the mechanisms that enable a shape shift. For a generic SSM, we consider the following three main components:

1. **Mechatronic design**, which defines actuators and latching mechanisms embedded in particles to enable shape shift, and also physical interfaces and mechanisms to support the spatial reconfiguration of particles. The mechatronic design is strongly influenced by the selection of a system architecture, and affects both scalability and particle dimensions, as Fig. 3.1 depicts.

2. **Power & Communication**, from an electrical perspective, defines how particles become the physical medium to support the networking and power supply (e.g., electrical grid in which particles are nodes). From a control perspective, it defines how the power is managed, for example, to save energy turning particle into low power mode. The communication among particles enable their cooperation, while

the communication between the ensemble of particles and external devices enables the control of the system. This calls for communication protocols to support, for example, routing, node addressing, and error correction. Fig. 3.1 indicates a vertical relationship between power & communication, system architecture and system scalability.

3. **Planning & Control**, corresponds to the software layer of the system which coordinates a shape shift from a global (planning) and local (control) perspective. Given a target shape, a plan is required to indicate how and when particles rearrange, taking into account the physical constraints imposed by the system architecture. The execution of the plan relies on control algorithms that supervise the local actuation of the particles. Fig. 3.1 shows the vertical relationship between planning & control, system architecture and system scalability.

As these three components are tightly bound to each other and all belong to the implementation layer, they are represented in Fig. 3.1 by means of nested blocks. In the next section, we closely analyze how the vertical relationships among requirements generate horizontal implications across requirements belonging to the the same abstraction level.

3.1.2 Shape-Shifting Display Requirement Analysis

After having derived the relationships between *system requirements* and *application requirements*, the next step is to examine how such system requirements can be implemented in *hardware and software*. Aware of the vertical dependencies that exist among distinct abstraction levels (Fig. 3.1), we derive the *horizontal* implications that cause conflicts among requirements at the same abstraction level.

The hierarchical diagram of Fig. 3.2 highlights the relevant aspects to be considered in the development of an SSD. Solid arrow lines interconnecting blocks indicate the *vertical* relationships earlier discussed among *application requirements*, *system requirements* and *implementation requirements*. The diagram is essentially an exploded view of Fig. 3.1. Dashed arrow lines indicate the *implicit relationship* among requirements. In particular, with respect to the four main types of architectures described in Sec. 2.2 and also summarized in the "Taxonomy" of Fig. 3.2, the diagram shows the impact that the choice of a system architecture has on the rest of the system. When solid and dashed arrow lines outline a loop, a conflict among the requirements involved in the loop has to be expected.

In the following we consider the three *system requirements* identified in the previous section and also reported in Fig. 3.2: I. System Architecture; II. System Scalability; III. Particle Miniaturization. We analyze their impact on the system, starting from the implications that derive from the selection of a specific system architecture. For each system requirement we consider its implications from an *implementation* perspective – i.e., mechatronic design, power & communication, planning & control – and also, when relevant, its implication from an *application* perspective.

I. **System Architecture**

One of the first and foremost questions on the design of an SSM concerns the choice of the underlying architecture. As the latter defines the way that particles interact with

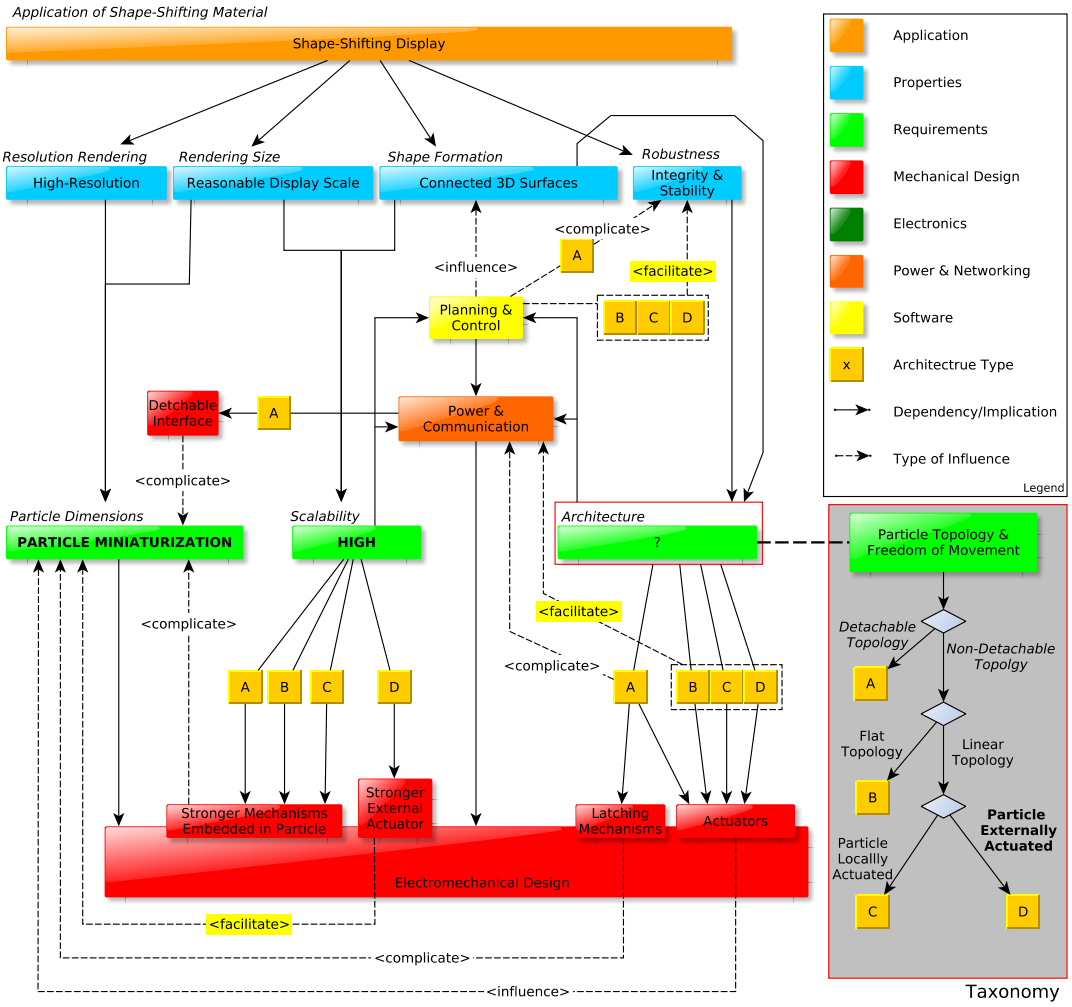


Figure 3.2: Requirements dependencies and comparison among architectures.

each other (i.e., particle interaction models) and their workspace (see Sec. 2.2.1), the architecture represents a fundamental decision that affects many aspects of an SSM. This includes primarily the ability to form arbitrary shapes, the mechanical robustness and complexity of the SSM, which in turn influences particle size and system scalability, namely display resolution and size. The *interdependence* among diverging aspects generates challenging situations, mainly because of the conflicts among requirements at the same abstraction level.

Mechatronic design. The complexity of the mechanical design strongly depends on the chosen architecture. As described in Ch. 2, we can identify four main architectures which are summarized in the "Taxonomy" of Fig. 3.2. These architectures determine particle's topology (i.e., bond and connectivity, as described in Sec. 2.2.1) and consequently the class of attainable shapes that an SSM can form. Architectures based on a detachable topology (A.) allow particles to detach from the ensemble

and spatially relocate (e.g., by climbing each other) to any vertex of a 2D or 3D *lattice*. This potentially enables the formation of virtually *any* shape. Conversely, non-detachable topologies like foldable *sheets* (B.) and *linear* structures (C. and D., and also including tree-type topologies), limit particle's freedom of movement to relative displacements among adjoining particles. Despite the fact that detachable topologies allow the formation of literally any shape, the complex architecture needed to enable particle migration, generally consisting of built-in actuators and latching mechanisms (see Fig. 3.2), complicates particle mechatronic design costs. Also, power supply and communication, planning and control become more challenging. Instead, the mechatronic design of non-detachable topologies (architecture B, C, D) is simpler, potentially more robust and facilitates other aspects like power supply and communication. For these reasons, our design approach focuses on non-detachable topologies.

Power Supply & Communication become more challenging in detachable topologies, because of particle mobility, from both hardware and software perspectives. Our assumption is that physically interconnected particles form a network, which can be exploited as an electrical grid to supply power to each particle and also to enable communication among particles for coordination and control purposes[1].

From a mechanical point of view, detachable architectures (A.) call for detachable electrical interfaces [75] or electromagnetic coupling [44], which complicate the mechatronic design. Also, specialized hardware is required to handle temporary power interruption, for example, in case it is not possible to supply to moving particles. From a software point of view, due to particle mobility communication protocols need to support mobile routing, particle localization, and also handle temporary network disconnection. Accurate power management is required to compensate for temporary disconnection from the electrical grid.

Sheets and linear structures (B., C., D.) can rely on permanent links among particles, which facilitate power supply and communication from a mechatronic perspective (no need for detachable electrical interfaces). Power management (e.g., low power mode) can be required to balance the load across the electrical grid. Particle localization might be required during the boot-up phase of the system, although it is possible assume that the location of a particle in a non-detachable topology is known *a priori*. The above reasons make us consider solutions based on non-detachable topologies to be suitable for the creation of an SSD.

Planning & Control algorithms are integral parts of an SSM, as the coordination of the ensemble of particles and their actuation depend on them. As Fig. 3.2 shows, a loop exists among architecture, planning and control, and shape formation. The formation of arbitrary shapes, indeed, is not a pure hardware design challenge but it also depends on the computational complexity necessary to plan a shape shift. In this regard, Hou et. al [53] demonstrate that the optimal reconfiguration problem is in class of complexity NP-complete for detachable tree-type and linear architectures, while for non-detachable topologies the question is still open. However, as the

[1] In this way, we exclude both the use of batteries embedded in particles (which would make miniaturization and maintenance quite difficult and expensive) and inconvenient electric cables to supply power to each individual particle.

demonstration proposed in [53] does not take into account all the physical constraints of the system, such as the maximum forces that the system can withstand, a higher complexity could exist.

Integrity & stability. The diagram of Fig. 3.2 shows that planning and control also influence the robustness of the system, despite its strict dependency on the system architecture. Planning has the goal to define a sequence of intermediate configurations necessary to carry out a shape shift, while control is responsible for executing the reconfiguration plan. Both planning and control need to take into account the physical constraints imposed by the architecture. In addition, an admissible planning strategy has to guarantee the actuation forces to be within a maximum range, for any intermediate configuration. Exceeding the maximum force that the SSD structure can bear, would undermine the integrity and the stability of the system.

II. **Scalability**

Scalable SSMs allow the number of robotic particles to increase without remarkable loss of performance and robustness. For an SSD, scalability is fundamental to support the visualization of reasonably sized shapes. Indeed, a large number of particles is required (Fig. 3.2), considering that high-resolution rendering implies tiny particle dimensions.

Mechatronic design. Scalability raises both mechanical and software challenges. In detachable topologies (A.) where the integrity and stability of the system rely on latching mechanisms providing adhesion among particles, scalability is challenging since a larger number of particles calls for stronger mechanisms, whose strength is limited by their size. Also in non-detachable topologies (B., C.) where the angular displacement between adjacent particles relies on actuators embedded in particles and acting at the hinges, the inherent relationship between size and strength of actuators limits the scalability of such systems and the possibility to withstand growing loads.

Externally actuated flat topologies (e.g., origami [36]) and linear non-detachable topologies [115], instead, consist of passive particles which are remotely actuated by an external actuator. As the external actuator can be potentially upgraded to support a larger number of particles, externally actuated architectures are in principle highly scalable. The main limitations of the two representative externally actuated systems describe in Sec.2.2.2, are that "inFORM" [36] requires a dedicated actuator for each particle (rising pins whose height is controlled to outline a dynamic surface), while "Ratchet14" [115] is a foldable chain that requires a dedicated manipulator (i.e., multiple chains require multiple actuators). This poses a limit to the scalability of these systems in terms of number of particles (inFORM) and of number of chains (Ratchet14).

Anther limitation we can observe, for example in Ratchet14, is that the maximum length of the chain depends on the forces that the passive latches embedded in particles can exert to retain a shape. To overcome this limitation, we propose a

solution that exploits the force generated by an external actuator for actuating the system and also for retaining the final shape.

Power & Communication. From a software perspective, scalability has implications on network protocols, energy management, planning and control algorithms. Network protocols need to ensure communication among a growing number of nodes, which makes protocols based on a shared medium – like bus or wireless system – not sufficiently scalable given the high density of nodes.

Planning & Control. As the computational complexity of planning algorithms depends, in the first place, on the number of particles, and then on the geometrical complexity of the target shape to form, planning algorithms need to scale with a growing number of particles. Although in general control defines the local interaction between adjoining particles, in some architectures, the complexity of control algorithm is a function of the size of the system. As explained in Chap. 4, we adopt a model-predictive control to compensate for the absence of sensors in particles. This solution reduces on the one hand the mechanical complexity of particles for higher scalability and resolution, but requires on the other hand a dynamic model of the system whose computational complexity depends on the number of particles.

III. **Particle Miniaturization**

Miniaturization essentially rises a design challenge for which actuators, latching mechanisms, power and communication interfaces need to be combined into one single autonomous device of reduced dimensions. As a matter of fact, a simple mechatronic design facilitates the subsequent manufacturing and miniaturization process.

Mechatronic design. As Fig. 3.2 depicts, particle miniaturization is a requirement for high-resolution rendering. In detachable topologies (A.) particles typically require embedded latching mechanisms and actuators to allow temporary disconnection from the ensemble and spatial relocation. This increases the mechanical complexity and hinders particle miniaturization.

Conversely, non-detachable topologies (B., C., D.) typically require actuators to regulate the displacement of adjoining particles, while latching mechanisms only provide the passive function of retaining a formed shaped. In particular, in architecture of type (D.) the main actuator is removed from particles, which are remotely actuated by an external actuator. This facilitates particle miniaturization.

Power & Communication complicates particle miniaturization in case of particle mobility (A.) has to be handled. Detachable electrical interfaces need to be embedded into particles, as well as auxiliary hardware to handle temporary power interruptions (e.g., capacitor or battery). In architectures (B., C., D.) the permanent links among particles simplify all the above aspects.

Planning & Control. Particle miniaturization affects planning and control as long as these two tasks rely on distributed algorithms, namely in case particles perform local computation to coordinate a shape shift. In this case, the main challenge lies in the limited computational capability of the microcontroller unit embeddable

in particle, whose physically dimensions are constrained. Despite the continuous increment of computational density [28], this raises a combined mechatronic and software design challenge.

3.1.3 Discussion

The selection of the underlying architecture entails many implications on multiple aspects of the system, often causing conflicts among requirements at different abstraction levels. Considering that detachable architectures complicate the fulfillment of the *system requirements* of an SSD (Fig. 3.2), namely scalability and particle miniaturization, we do not consider this type of architectures for the realization of an SSD.

Non-detachable architectures (B., C., D.) in general facilitate, or at least do not hamper, the fulfillment of *system requirements*. In particular, externally actuated architectures (D.), which rely on an external actuator to remotely actuate particles, facilitate system scalability and particle miniaturization as the external actuator can be upgraded without affecting the particle design. These characteristics make such architectures compliant with the *application requirements*, in particular, towards the realization of a high-resolution rendering SSD. In the following, we explain our design approach to externally actuate the shape-shifting surface of an SSD. This allows us to combine a large number of tiny particles for high-resolution rendering.

3.2 Curvature-Controllable Chains

In this section we observe that a Shape-Shifting Display (SSD) is essentially a surface that can be decomposed into piecewise foldable chains, with the latter consisting of concatenated programmable particles. Inspired by existing solutions that show the advantage of having externally actuated chains, we devise a mechanical solution to remotely actuate multiple chains by means of a single external actuator. In this setup, particles locally control the curvature of the chain by exploiting the externally provided force.

In the following, we first explain how we intend to design an SSD using foldable chains, which are simple to build and to actuate by means of an external actuator. We base our SSD design on the concept of "Force-Guiding Principle", which is inspired by the combination of two related works presented in this section. We design foldable chains to verify the applicability of the "Force-Guiding Principle". Following the methodology indicated in Sec. 1.5, a progressive refinement of the initial design is obtained through experimental validation. As conclusion of this section, we discuss the potential scalability that derives from the application of the "Force-Guiding Principle".

3.2.1 Multi-Chain Surface

Despite the fact that detachable topologies allow the formation of literally any shape, the complex architecture necessary to enable particle migration (generally consisting of built-in actuators and latching mechanisms) raises costs, limits the scalability of the whole system, and hampers particle miniaturization and networking (i.e., power supply and communication). The formation of arbitrarily complex shapes, however, does not necessarily

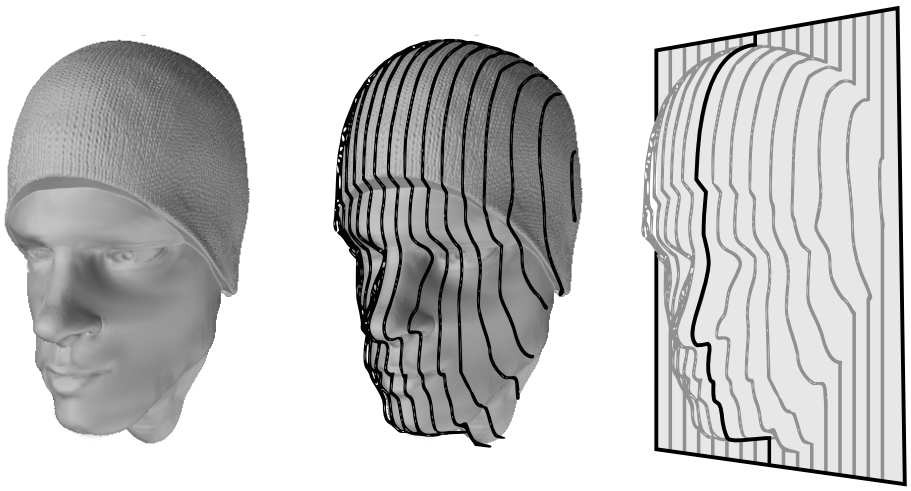


Figure 3.3: A Shape-Shifting Display can be decomposed into multiple piecewise foldable chains, where each chain outlines a slice of the target shape to display. A chain results from the concatenation of robotic particles responsible for the local curvature.

require sophisticate architectures. We observe that a Shape-Shifting Display (SSD) is essentially a shape-shifting surface that can be built starting from a foldable flat topology or a linear topology. The advantage of adopting a linear topology is that external actuation is possible, and thus system scalability is facilitated as described in the previous section.

As Fig. 3.3 illustrates, an SSD can be built upon the combination of piecewise foldable chains regularly arranged to be parallel to each other, and by having each chain outline the contour of a slice of the target object. A chain is in turn composed of robotic particles, which control its local curvature by modifying their relative *angular* displacement. To be suitable for an SSD, a chain requires particles with at least two Degrees of Freedom (DOFs) in order to obtain a 2D discrete workspace[2].

As discussed in the previous section, a chain topology improves system robustness and simplifies particle mechanical design. This result in more lightweight and cost-effective solution. However, actuators embedded in particles limit the system scalability, due to the leverage effects which entail an upper-bound on the maximum length of the chain. Fig. 3.4 shows our first prototype consisting in a concatenation of modular servomotors (Dynamixel), which form a piecewise foldable chain (to some extent, similar to the Millimoteins [65] presented in the previous chapter). Each element is powered at a constant voltage, supplied through a common bus deployed along the chain. The bus also supports serial communication (RS245 standard) to remotely control each servomotor. Beside the advantage of being simple to assemble, the prototype demonstrates that the solution is hardly scalable. Considering that the actuators embedded in the head of the chain are subject to the weight and inertia of the tail of the chain, a dozen of elements are sufficient to make the system unstable. With many elements also the supplied voltage drops along the bus, which causes undesired reset of remote nodes. A servomotor embeds a control unit to handle actuation and communication, auxiliary embedded logic, an electric motor

[2]The workspace indicates the spatial locations a particle can eventually reach after a shape shift, as defined in Sec. 2.2.1

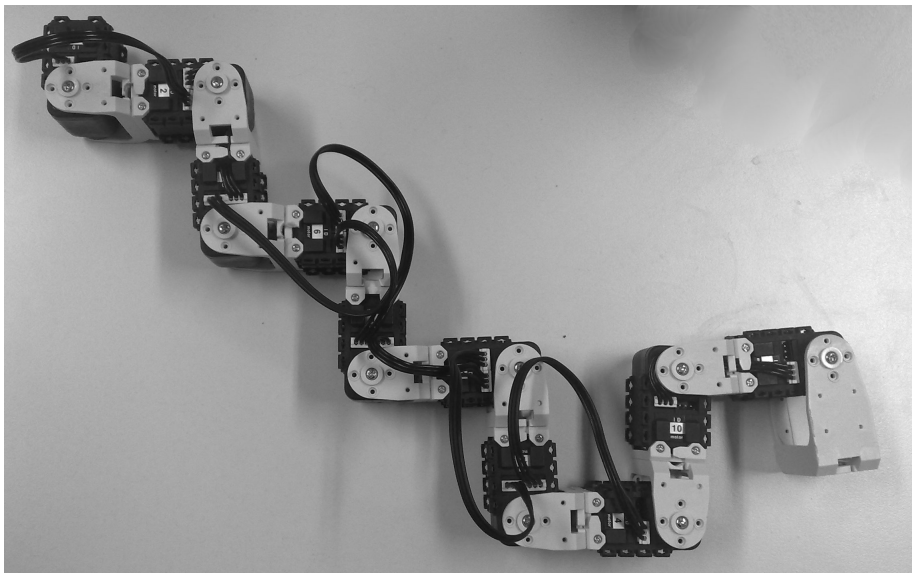


Figure 3.4: First Prototype: using Dynamixel® [1].

and reduction gears. All these components increase the complexity of a node and the cost of the system.

To overcome these limitations, our solution relies on an external actuator to remotely actuate particles and thus minimize their mechanical design, with an approach similar to [111,115]. As a further step, we devise a mechanical solution to decouple the actuation among different chains and exploit a unique actuator for the whole SSD. In the proposed solution, the external actuator is also responsible for retaining the target shape, without the need of extra latching mechanisms. This also contributes to reduce the mechanical complexity of the system.

3.2.2 Inspiring Related Work

The combination of the two works described below and also mentioned in Sec. 2.2.3 have inspired our first design of an externally-actuated foldable chain.

Square-Tile Chain. Griffith [46] demonstrates that a 2D foldable chain can approximate 2D shapes. Based on modular square tiles sequentially hinged at opposite vertices along a diagonal (Fig. 3.5), consecutive tiles modify the curvature of the chain by rotating about the common hinge until two edges touch each other. When free from external forces, a chain tends to fold under the effect of magnetic forces due to permanent magnets embedded in each tile, which cause consecutive tiles to fold on a predefined side. As to program the formation of a shape, magnets need to be manually installed in tiles, the proposed solution is not suitable for the creation of an SSD where chains need to dynamically adapt to new configurations in reasonable time and possibly through a software controlled approach. Nevertheless, our first prototype grounds on a similar principle where, instead of hard-coding the target shape, a selector embedded in tiles determines the folding side, while an external force actuates the chain.

Externally-Actuated Chain. Starting from the consideration that removing the main

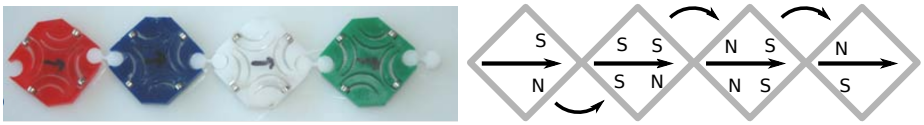


Figure 3.5: Prototype of foldable square-tile chain [46] and its working principle.

actuation from the robotic particles higher scalability and particle miniaturization can be achieved, White at al. [111, 115] devise an external manipulator to actuate a foldable 3D chain (Fig. 2.2l). They show that a chain composed of concatenated right angle tetrahedrons can outline arbitrary 3D curves when externally actuated. To retain a formed shape they propose two solutions based on either permanent magnets [115] or mechanical latches [111], embedded into particles. In order to unfold the chain and restore the initial straight configuration, manual actuation is adopted in both cases. In particular, in [111] a Shape-Memory Alloy (SMA) spring unlocks the mechanical latch to release the chain under the effect of gravity. To activate the unlocking system a heat source is manually provided by means of a heat gun.

Although the proposed working principle has the potential to free particles from bulky built-in actuators, thereby facilitating the creation of a high-resolution SSD, the approach is not suitable for our application. Considering indeed that a chain requires a dedicated external manipulator, the solution is not applicable to an SSD where the scalability in orthogonal directions calls for multiple long chains, and for which only a single external actuator should be devised for the whole system. The issue lies in the fact that the kinematics of the external manipulator and of the chain are tightly coupled. Indeed, in [111], the motion of the manipulator that is required to fold the N-th particle, needs to take into account the current folding state of the other $N-1$ particles. Consequently, as actuator and chain cannot be "decoupled", it is not possible to allow a single manipulator to actuate multiple chains simultaneously.

3.2.3 Force-Guiding Principle

The inherent relationship between size and strength of actuators complicates particle miniaturization and system scalability. To overcome this limitation, our solution consists in adopting an *externally-actuated chain* approach, by removing the main actuators from within particles and by outsourcing the generation of actuation force to an external actuator. As the latter can be upgraded to support a larger number of particles, the system is eventually scalable.

For an SSD composed of multiple chains (Fig. 3.3) not only the number of particles per chain needs to scale up, but also the number of chains. Considering indeed that the length and number of chains determine the size of the visible area of the SSD, this is required in order to obtain a *high-resolution rendering* SSD with a *reasonable display scale* (according to the application requirements discussed in Sec. 3.1).

White at al. demonstrate the feasibility of an *externally-actuated chain* approach in [111, 115]. However, their solution requires a dedicated manipulator per chain as particles are completely passive elements only subject to the action of the external manipulator. In this way, the curvature of the chain does not depend on the local action of particles, but is only determined by the kinematics of the external manipulator. In other words,

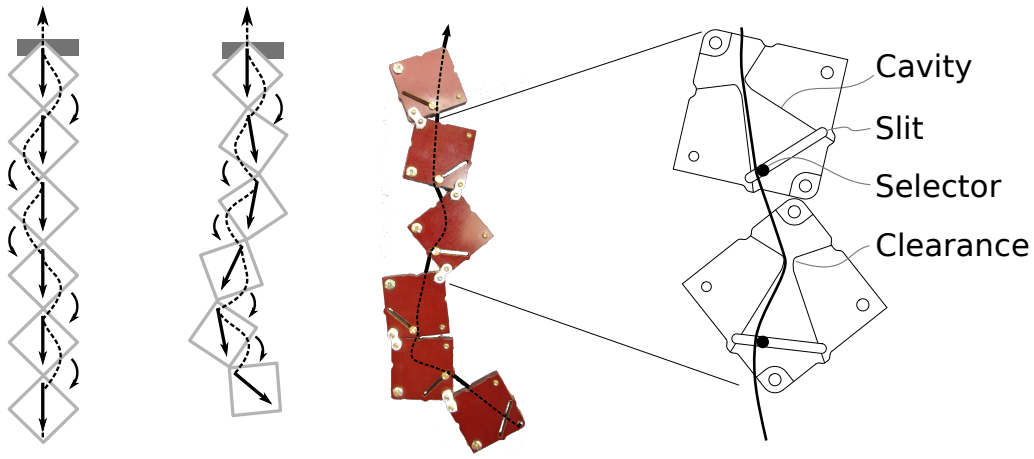


Figure 3.6: Force-Guiding Principle applied to a square tile chain. Tiles are hinged at the two corners along a diagonal. Consecutive tiles can fold on one of the two sides in order to outline a target shape. Each tile embeds a selector to indicate the folding side of the tile with respect to the following one. A tendon passing through the tile chain exerts a tensile force necessary to actuate the system. This second prototype is manually actuated.

not only does the external manipulator generate the forces to actuate the chain, it also selects on which side each particle eventually folds. This entails a tight coupling between the external manipulator and the chain, which prevents a single manipulator from being used to actuate multiple chains simultaneously. Considering the mechanical complexity, size and costs of the external manipulator, it is not possible to scale up the number of chains by simply replicating the external manipulator.

Our hypothesis is that a tendon-driven solution can be adopted to decouple the external actuator from the chain by letting each particle locally select the folding side. In this way, a tendon deployed along the chain propagate a compressing force that act homogeneously on each particle and that is independent of the target folding side of the particle. To locally control the curvature of the chain, each particle "guides" the externally provided force to fold the chain on the desired side by means of a built-in tiny mechanism. As the external actuator is only responsible for generating the actuation forces (i.e., pulling the tendons), its kinematics is decoupled from chain and multiple chains can be simultaneously actuated by the same external actuator. We call this approach "Force-Guiding Principle".

Considering that a minimum tension applied to the tendon is sufficient to retain the shape, no latching mechanisms are required to be embedded into particles. In addition, assuming that the force-guiding mechanism embedded in particles can be designed to be very minimalistic, particles are amenable for miniaturization and the system is scalable.

3.3 Tendon-Driven Square-Tile Chain

Our first attempt to demonstrate the feasibility of the Force-Guiding Principle takes inspiration from the work that Griffith et al. proposed in [46]. As depicted in Fig. 3.5, a 2D chain is composed of square-tile particles interconnected at the opposite vertices along a diagonal. Built-in magnets force each pair of consecutive particles to fold on one side. This

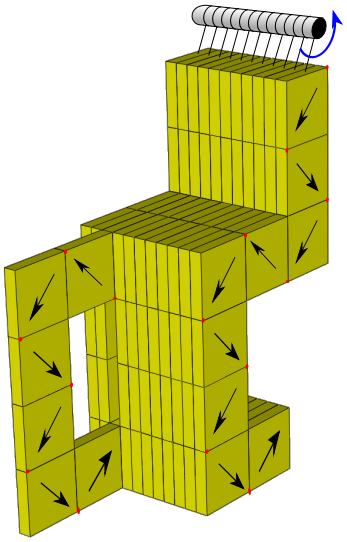


Figure 3.7: 3D Surface: multiple square tile chains are combine together to form a programmable surface. An external actuator actuates all the tendons embedded in chains at the same time.

solution shows that a 2D chain can form arbitrary 2D shapes, despite its simple design. With a similar approach, we adopt a square tile chain to outline the contour of 2D shapes, i.e., a slice of the target 3D object to display. However, in order to improve the scalability of the chain, we implement our Force-Guiding Principle to allow the simultaneous actuation of multiple chains and enhance the scalability of the chain.

3.3.1 Mechatronic Design

In order to build a Shape-Shifting Display (SSD) using curvature-controllable 2D chains, chains are arranged parallel to each other and with their workspace perpendicular to the flat surface of the display (when the display is at rest position). We also assume that all chains have their head particle firmly fastened to the top bracket of the SSD and that unactuated chains can freely swing parallel to each other without reciprocally interfering their movements.

Aiming at the realization of a scalable particle chain, we propose a solution to apply the Force-Guiding Principle to the square-tile chain presented in [21, 46] and overcome scalability limitation that derive from the use of built-in magnets.

Fig. 3.6 shows the prototype of a square tile chain where an externally actuated tendon traverses the chain. The tendon is tied to the tail particle of the chain and pulled from the top particle, which is steady. In this way, a tension applied to the tendon causes a compressing force on the chain and its particles. As a chain forming an SSD is hinged by one extreme to a top bracket of the SSD, the gravitational force causes its particles to be initially vertically aligned. A slight misalignment between the tendon and any of the hinges interconnecting consecutive particles, cases the two consecutive particle to fold under the effect of the compressing force.

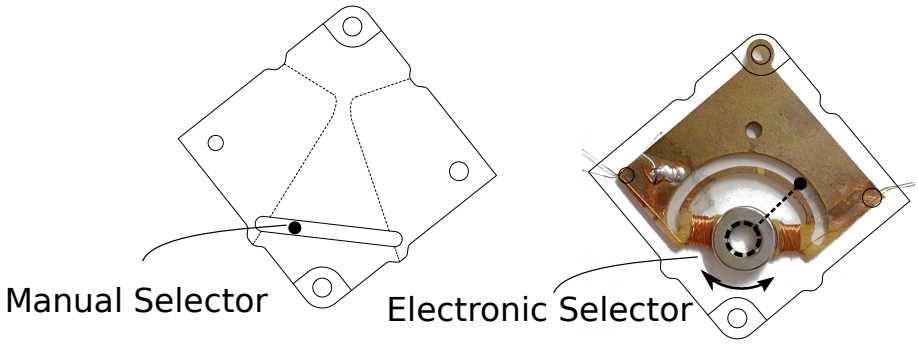


Figure 3.8: Attempt to build a remotely controllable selector.

In order to control the folding side of consecutive particles – and hence to define the shape that the chain eventually outlines – we control the path of the tendon traversing the chain by means of a sliding selector. Assuming that the tendon *enters* a particle from the top and *leaves* it from the bottom (Fig. 3.6), the selector deviates the path of the tendon leaving the particle towards one of the two folding sides. The selector consists essentially in a nut and bolt fastened across a slit, which is machined in the body of a particle, as shown in Fig. 3.6. As the selector alone is not sufficient to make the system controllable, a narrow clearance positioned along the diagonal of the square tile, near to upper corner, constrains the path of the tendon *entering* the particle, as the sectioned tiles of Fig. 3.6 show. The adopted mechanical expedient induces a leverage effect between two consecutive particles which consequently fold. In this way, the shape that a chain outlines is controllable and depends on the configuration of the sliding selectors. With multiple such chains actuate by single actuator, we can build the 3D surface of Fig. 3.7.

Manual Folding-Side Selection. As a proof of concept, the prototype shown in Fig. 3.6 is built for manual actuation. The operator configures the folding side of each particle by changing the position of the sliding selector. The tendon is placed between the sliding selector and the target folding side of the particle. For example, the top particle of the chain in Fig. 3.6 folds right, therefore the selector is set towards the right side and the tendon is positioned to the right of the selector. It is worth to notice that the selector of the second particle has no effect on the first particle, due to the small clearance that constrains the tendon to pass near the top corner of the second particle. Once all particles are configured, the operator holds the head of the chain (top particle in Fig. 3.6) and pulls the tendon to actuate the chain, until all particles are fully folded. Particles start folding from the tail of the chain and eventually the chain outlines the target shape. To retain the final shape, it is sufficient to apply a minimum tension on the tendon.

Remote Folding-Side Selection. In order to make a square-tile chain suitable for the construction of an SSD, the chain should be remotely controllable. This requires an electronic selector to be embedded in each particle and that can be remotely operated to configure particle's folding side. An attempt to built such a selector is shown in Fig. 3.8. An elementary bistable motor is obtained combining two solenoids and a cylindrical permanent magnet diametrically polarized. The solenoids are wound around the prolonged arm of the thin board shown in Fig. 3.8 that work as stator of the motor. The hollow permanent magnet placed in the middle is free to rotate and works as rotor. A short arm

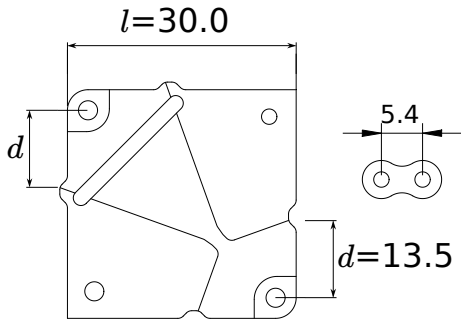


Figure 3.9: Dimension of the components of a square-tile chain.

attached to the rotor carries the tendon along the curved slit. When the tendon is forced to move towards one of the two extremes of the slit, a small recess in the geometry of the slit prevents the selector from changing its position. The realization of this device has shown its inability to perform the expected task, due to the small force that the motor can exert. Considering that the electromagnetic force is proportional to the number of coils and that the existing setup counts about 50 coils, which are not sufficient to cause a rotation of the motor only subject to the friction with the surface it lies on, the approach was abandoned.

However, this underlines the importance of the particle design, which is fundamental to successfully implement the Force-Guiding Principle. The lesson learnt inspires the design ultimately chosen, which is described in Sec.3.4.

External Actuation. Assuming that the folding side of particles in the chain can be remotely configured, the chains composing an SSD can be actuated by a single external actuator, as Fig. 3.7 depicts. A shaft connected to the external actuator (not shown in the picture) winds up the tendons of the regularly arranged chains. We assume particles in chains to be configured and simultaneously actuated. Despite the fact that chains are independent from each other, a physical constraint imposes an equivalent number of particles per chain, and also particles having all the same size.

3.3.2 Manufacturing

The prototype shown in Fig. 3.6 is composed of square tiles interconnected at their hinges by means of "8"-shaped plastic links. Fig. 3.9 indicates the dimensions of the components. The tile body is composed of two equivalent thin layers combined into a sandwich and fastened by means of two screws at the corners. Using a CNC machine each layer is precisely cut and cavity and slit machined on the surface. The material in use is "fabric-base laminate" with a raw thickness of 3 mm. It costs about $100 €/m^2$.

3.3.3 Shape-Shifting

The square-tile chain allows consecutive particles to fold on either sides – i.e., left or right. Given the contour S of a target shape, in order for a chain to outline the desired shape, a folding configuration has to be inferred. Particles of completely folded chains lie on a square grid [46] as shown in Fig. 3.10. Each feasible configuration corresponds to a

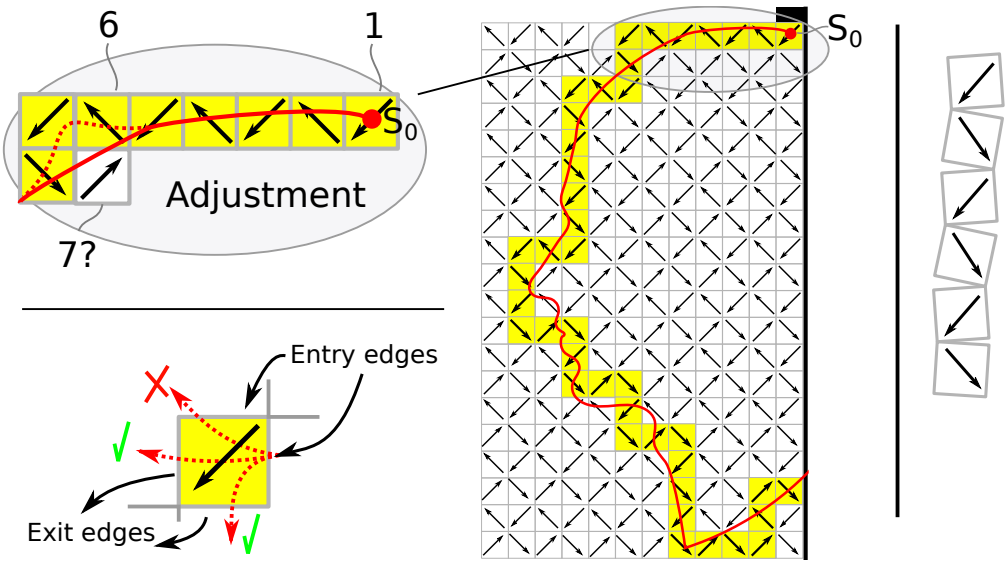


Figure 3.10: The arrow pattern on the square grid is generated considering all the possible folding configuration of a chain. Arrows indicate the position of the hinges interconnecting consecutive particles.

sequence of tiles on such a square grid. The superposition of the target contour S onto the grid indicates a sequence of particles from which the folding configuration can be deduced.

As the physical constraints of the chain determine the set of possible folding configurations, the set of contours that the chain can outline is limited. The arrows depicted in Fig. 3.10 are aligned to the hinges interconnecting consecutive particles. Only contours that traverse each particle in the same direction indicated by the arrows can be outlined by a square-tile chain whose configuration derives from the sequence of traversed particle.

Let us assume that the particle in the top-right corner of the square grid is the head particle of the chain (i.e., the one fixed to the SSD bracket). The next particle can only lie at the two positions West and South, namely next to the edges forming the corner which is pointed at by the arrow. By iterating the reasoning to any other tile on the grid (i.e., considering each possible folding configuration), we observe that once the orientation of the head particle is defined, the orientation of any other particle is *univocally* defined.

A contour S can only be outlined by a chain, if, for a chosen walking direction along S, S leaves each particle it traverses on the grid from one of the two edges next to the arrow head. This implies also that the contour S enters the next particle on the grid from one of the two edges next to the arrow tail. These two conditions reflect the physical constraints of the chain.

In order to deduce the folding configuration to outline the contour S, the following algorithm can be applied. Starting from the extreme S_0 that lies within the head particle of the chain and walking along the contour S, we observe the sequence of particles that S traverses. As long as S leaves each particle it traverses from one of the two edges next to the arrow head, the sequence of traversed particles is valid. Instead, when this condition is not satisfied, the path needs to be *adjusted* through approximation. Fig. 3.10 shows this situation. The profile head touches and correctly traverses the first six particles (zoomed

grid). Instead, the seventh particle is traversed in a wrong direction, because S enters the particle from an edge next to the arrow head of particle 7. Therefore, the path needs adjustment as the dashed red line in Fig. 3.10 indicates. This solution allows a square-tile chain to approximate the target profile and satisfies the physical constraints.

3.3.4 Experimental Validation

We build and manually actuate the chain, as earlier described. The chain behaves as expected, although the friction between tendon and folding selectors makes it difficult to fold too articulated configurations. A significant limitation of the mechanical design, derive from the 8-shaped link interconnecting particles that cannot guarantee a proper alignment of consecutive particles. This occasionally causes inappropriate chain layouts, which prevents the chain from being completely actuated and folded. Consequently, not all the configurations that are in principle feasible, are in practice attainable with the prototype at hand.

The worst case scenario is identified for the chain outlining a cantilever. Although we do not have available measurements of the required actuation forces for the worst case scenario, we observe a significant increase of such forces when manually actuated. The main reason is attributed to the modest leverage effect that the combination of folding selector and cavity clearance produces on adjoining particles. A possible solution to improve the lever is to increase the value of d (see Fig. 3.9), namely the distance between the hinge (pivot point) and the application point of the tensile force. However, this solution improves the lever only when particles are partially folded, and has no significant effect when the chain hangs straight.

The friction between tendon and the folding selectors make the tendon wear out rapidly till its breakage after about twenty actuation cycles. This is another consequence of the reduced lever, which increases the overall mechanical stress of the tendon.

3.4 Tendon-Driven Triangular-Particle Chain

In this section we refine the chain design proposed in the previous section, in order to demonstrate the feasibility of the Force-Guiding Principle and the possibility to build a remotely controllable chain. As we notice that the main limitation of the square-tile chain is the scarce leverage effect between consecutive folding particles, we demonstrate that a different particle geometry overcomes such a limitation. Also, by means of a pair of counteracting tendons (instead of a single tendon as in the previous design) we maximize the leverage effect, and thus the force that the external actuator needs to exert on the remotely actuated particles. In addition, the new design allows us to embed simple actuators in each particle to remotely control the folding side of the particle. This is a fundamental requirement towards the realization of a Shape-Shifting Display (SSD) composed of foldable chains.

This section summarizes and extends concepts and results presented in Paper $\boxed{A}$.

3.4.1 Mechatronic Design

Similar to the design presented in Sec. 3.2, the workspace of each chain is perpendicular to the flat surface of the display (when the display is at rest position). In this way, chains work parallel to each other and can swing without mutual interference. Also, we assume that chains hang from a bracket top of the SSD to which the head particle is firmly fastened.

Aiming at the realization of scalable particle chains, we overcome the limitation imposed by the earlier presented design, with a solution that applies the Force-Guiding Principle to a triangular-particle chain.

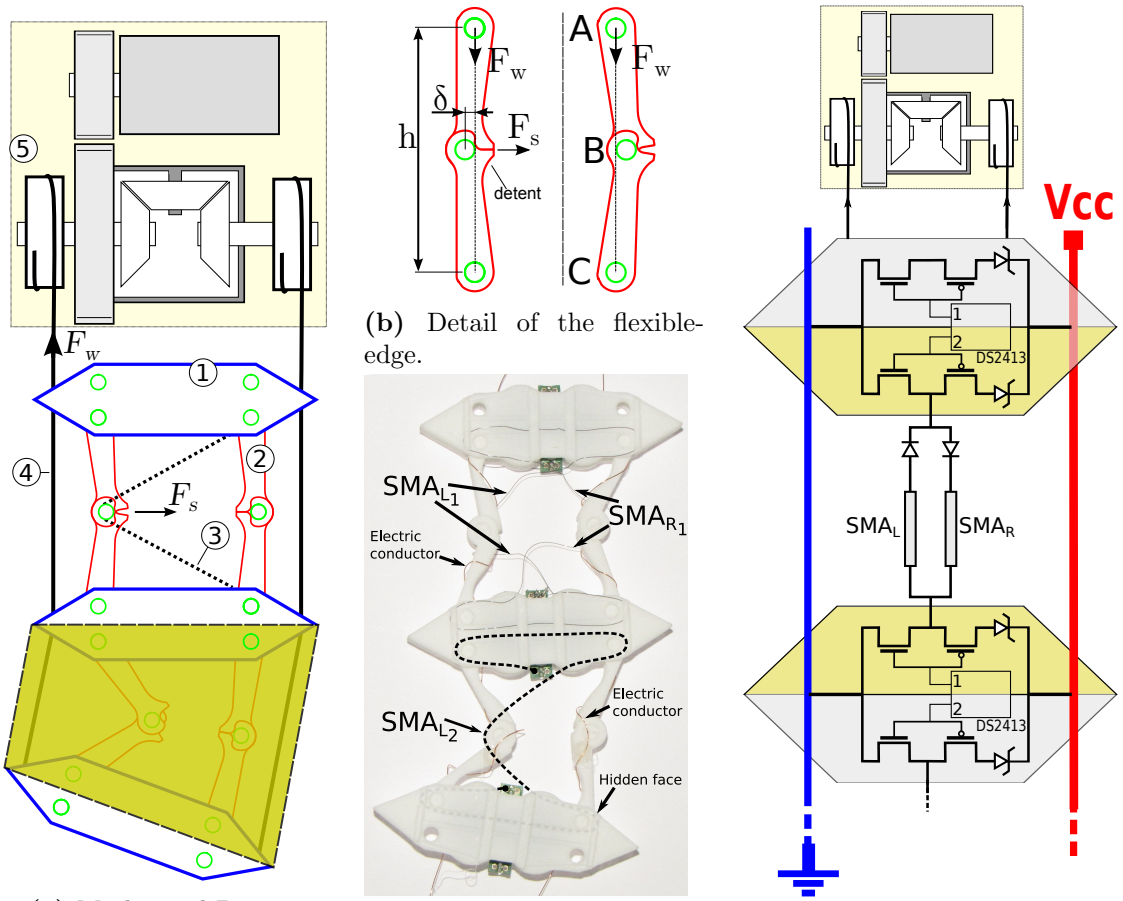
The key aspect is that particles forming a piecewise foldable chain can locally control the curvature of the chain, actuated by an external force. An advantage of the earlier described prototype is that only a single tendon is required to actuate the chain, which allows multiple chains forming an SSD to be actuated by a single external actuator. However, the presented solution significantly limits the maximum admissible load that the chain can lift (e.g., cantilever configuration), because of the modest leverage effect that the tendon produces on consecutive folding particles. This limitation is a consequence of the chosen square geometry and of the relative position of the hinges. As it is not possible to improve this condition by modifying parameters in the previous design, a different design approach is proposed in this section.

Mechanics: Force-Guiding Principle

Our solution, summarized in Fig. 3.11a, consists of particles initially outlining a rectangular shape that fold into an equilateral triangle, and by that modify the curvature of the chain. A chain is fully configured, and hence outlines the target shape, when all particles in the chain are folded into a triangle.

The chain presents an articulated structure that resembles a ladder, where a pair of tendons (element (4.) in Fig. 3.11a) are deployed alongside the chain and exert a compressing force on each particle. The body of a particle consists of six rigid links, which are sequentially interconnected at their extremes through revolute joints [80] and form a loop. We distinguish between the upper and the bottom base of a particle, which correspond to the long links (1.) indicated in Fig. 3.11a, and the lateral edges (2.). The upper base of the head particle of the chain is fixed to a bracket, top of the SSD. Consecutive particles in the chain have their bottom and top bases coincident.

Tendons traversing the chain alongside are tied to the bottom base of the tail particle and pulled upwards. This results in a tensile force compressing all particles in the chain. The Force-Guiding Principle is applied to exploit such a compressing force to fold the chain: by unlocking one of the two lateral edges symmetrically arranged at opposite sides of each particle, the compressing force folds the particle and thereby the chain. In particular, as Fig. 3.11b details, a monostable mechanism is obtained through a slight misalignment δ among the three joints A, B, C combined with a detent that constrains the relative rotation of the two links (AB and BC) forming the edge. In its straight configuration a lateral edge "locks" the layout of the particle in the initial rectangular layout (for example, the upper particle of Fig. 3.11a). To "unlock" the particle and thus fold the chain, a weak force applied to the middle joint (B) of the lateral edge, moves the middle joint towards the center of the particle. When the middle joint moves past the line between the other



(a) Mechanical Design.

(b) Detail of the flexible-edge.

(c) SMA path extension.

(d) Electrical block diagram.

Figure 3.11: Mechanical components of the system.

two extremes the particle is unlocked and the external compressing force folds it. This realizes the Force-Guiding Principle.

To unlock the lateral edge, two Shape-Memory Alloy (SMA) wires (3.) exert the force F_s (shown in Fig. 3.11a) necessary to unlock an edge. The mechanical displacement that a SMA wire can produce is relatively small compared to the displacement δ required to unlock an edge. This is due to the reduced shrinking ratio of the wire, which is 4% of its total length. In order to improve the overall displacement, we extend the length of the SMA wire by arranging it through tiny pipes obtained in the top and bottom bases of each particle, as Fig. 3.11c depicts.

Electronics: Power and Communication

An electric current is applied to the SMA wire, in order to unlock the lateral edge of a particle. This is a consequence of the joule effect that, by heating up the wire, make it shrinks. The electric power is supplied to particles through a pair of conductors deployed alongside the chain in proximity of the lateral edges. An electrical block diagram is shown in Fig. 3.11d.

As the two SMA wires actuating a particle work in mutual exclusion, two anti-parallel diodes (Fig. 3.11d) allow the use of a single H-bridge for both wires. By inverting the flow of the applied electric current, each SMA wire can individually activated. As the extremes of an SMA wire are located in the top and bottom bases of the particle (in order to increase its length, thereby its shrinking effect), two half-H bridges embedded in each base control the direction of the current applied to an SMA wires, as Fig. 3.11d depicts. Considering that when activated an SMA wire draws about $300\,mA$, it is not possible to actuate all the particles in a chain simultaneously. This entails a physical constraint to be taken into account at the *planning and control* abstraction level.

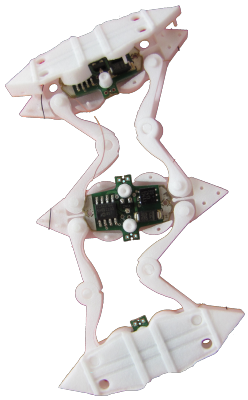
Given that the external actuator and particles need to actuate simultaneously to properly fold the chain, a *1-Wire® communication protocol* is adopted to establish a synchronous communication between the external actuator and the particles. A relevant characteristic of this protocol is that it can share the same pair of conductors used for power supply. A Dual-Channel Adressable Switch (DS2413, shown in Fig. 3.11d) is embedded in each particle and remotely controlled via the 1-Wire®protocol. As the current required to actuate the SMA wires is higher than the current that an addressable switch can withstand, an H-Bridge is required to drive the supplied current. The circuit board of a particle is shown in Fig. 3.12a.

Mechanics: Multi-Chain SSD

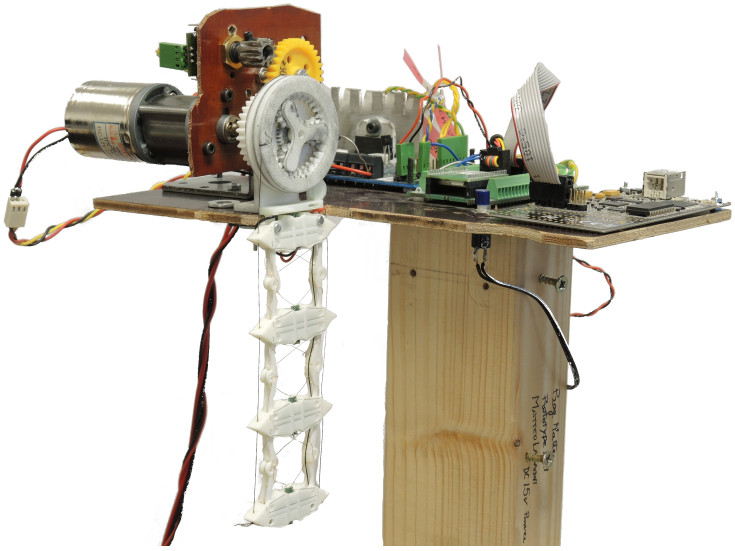
As each chain outlines a specific slice of the target 3D model, it is unlikely that all chains eventually have the same final configuration. This requires the actuation of multiple chains to be decoupled, in order to a single actuator for the whole SSD (Fig. 3.12b). An advantage of the tile-square-particle chain is that the presence of a single tendon allows a drive shaft to simultaneously actuate multiple chains by uniformly winding up all the tendons. Conversely, the two tendons of a triangular-particle chain complicate this operation, due to their asymmetric motion: when a particle in the chain folds on one side the corresponding tendon shortens, while the other tendon slightly extends.

To overcome this asymmetric behavior, our mechanical solution exploits a *differential winding mechanism* (element (5.) in Fig. 3.11a) integrated in each chain, to balance the tensile forces between the two tendons and to compensate for their asymmetric motion. Such a mechanism is essentially a differential gear placed in between the drive shaft and the two winches which wind up the tendons. Drive shaft and winches are designed to rotate in the same winding direction (an exploded view of the chain in Fig. 3.18 on page 61 shows this fact). As long as the forces between the two tendons are equivalent, the tendons are wound up at the same rate (i.e., their linear motion is equivalent). Instead, when the tensions between the two tendons diverge, the differential winding mechanism compensate for this asymmetry by inverting the relative rotation of one of the two winches: the winch connected to the tendon with the higher tension decreases its velocity (or even inverts it), while the other winch increases its velocity (Fig. 3.11a). This allows a single drive shaft with a unidirectional rotation to actuate multiple chains, as the differential winding mechanism integrated in each chain decouples the actuation of the chains.

This enables the realization of an SSD composed of multiple triangular-particle chains that rely on a single external actuator, with the constraint that chains must all have an equivalent number of particles, and the latter must all have the same size. In addition,



(a) A chain of two particles. The electronics embedded in each edge includes an Addressable Switch and a two half H-Bridge.



(b) Prototype of a SSD with a 3-particle chain.

Figure 3.12: Chain prototype and SSD prototype.

as it is not possible to fold all the particles of an SSD simultaneously, because of the high electric current required to unlock a lateral edge, only a subset of particles can be folded at the same time with the constraint that the same number of particles is folded in each chain. This constraint is solved at the *planning and control* abstraction level, further discussed in the next chapter.

3.4.2 Shape-Shifting Strategy: Pre-Planning Algorithm

In an SSD each chain approximates the contour of a slice of the target 3D object to be displayed. As multiple chains fold simultaneously, planning algorithms need to make sure that an equivalent number of particles is folding among all chains. The planning process consists of two phases: 1. a "Pre-Planning" phase to determine the *final configuration* of each particle in the chain given the target shape to be outlined; 2. the application of a planning algorithm, which takes into account the physical constraints and the dynamics of the system and infers a sequence of intermediate configurations to progressively reach the target configuration. In the remainder of this section, we focus on the pre-planning algorithm, as the complete planning process is discussed in the next chapter.

A chain configuration is a sorted sequence of binary values that indicate for each particle in a chain the folding side – i.e., left or right. Given a target contour S, the "Pre-Planning"algorithm computes the chain configuration such that the folded chain eventually approximates the contour S. If we consider all the possible configurations of a chain, assuming the particle head of the chain to be fixed, we observe that the particles of completely folded chains generate a triangular grid, as shown in Fig. 3.13. By overlapping the contour S on such a grid, in a way that one extreme of S coincide with the particle head of the chain (i.e., the one fixed to the bracket indicated by a black block), we can infer the configuration of the chain.

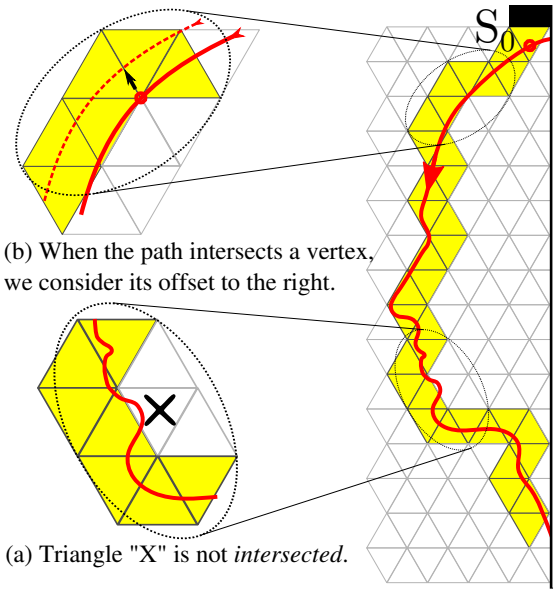


Figure 3.13: For each possible configuration of a chain, particles lie on a triangular grid.

Starting from the top-most extreme S_0 of S, we identify the sequence of triangles that S traverses, applying the following rules:

(a) Only those triangles that S enters through an edge and then leaves through a different edge are considered as valid. For example, the triangle marked with an "X" in Fig. 3.13(b) is not considered to be valid, because S enters and leaved the triangle "X" through the same edge.

(b) If the path S intersects a vertex on the grid, we consider a slight offset of the path to one side, as shown in Fig. 3.13(a).

The identified sequence of triangles maps to the sequence of particles in the chain. The two edges of a triangle correspond to the top and bottom edges of a folded particle. As S originates from the head particle of the chain, S enters each triangle through the edge that corresponds to the top edge of the particle. In this way, considering the relative position of the non-intersected edge and S, we can infer whether a particle folds left or right. Following S starting from S_0 once more, if the non-intersected edge lies on the left side of S, the particle folds left; if it lies on the right, the particles folds right. This process is applied to all the chains of an SSD to infer the configuration of the whole SSD.

Comparison of the square tile and triangular particle approaches.
We compare the proposed square-tile chain and triangular chain to demonstrate that the latter facilitates the determination of a folding sequence, and imposes fewer psychical constraints than the former approach.

Let us indicate with the capital letters A and V the triangles of a triangular grid having vertices pointing upwards and downwards respectively, as shown in Fig. 3.14. Assuming that S is a non-self-intersecting curve overlaid on the grid, we notice that a folding sequence is valid only if the contour S traverses alternately particles of type A and of type V (e.g.,

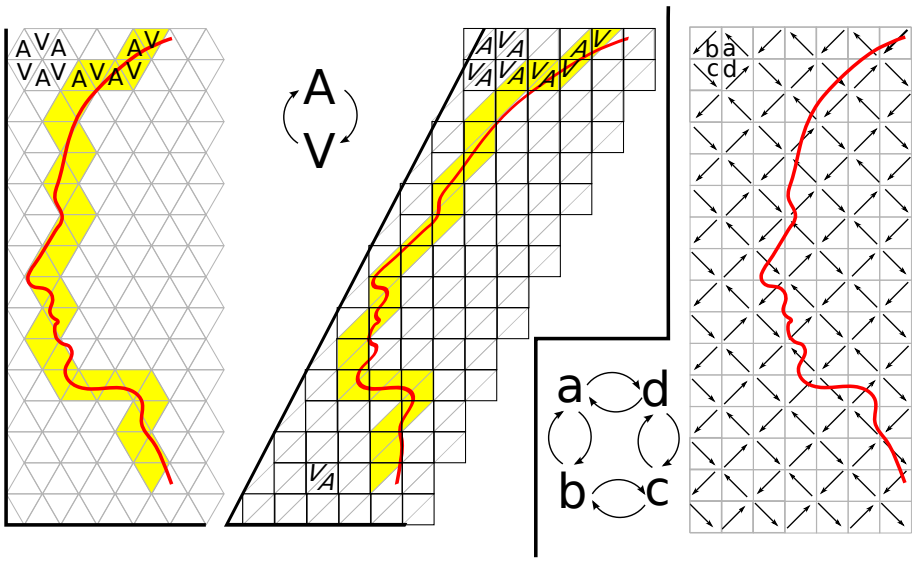


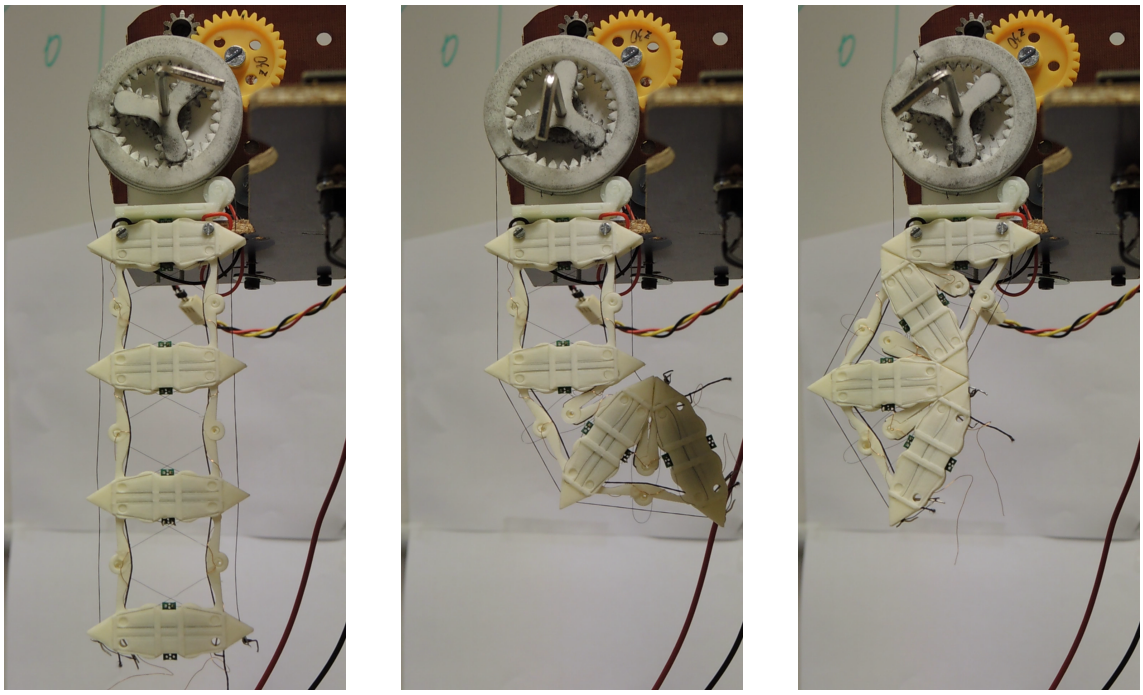
Figure 3.14: Geometrical properties of triangular and square tile chains.

AVAV is a valid folding sequence, while AVAAV is not).
Similarly, we indicate with the letters a, b, c, d the tiles of a square grid, where each letter is associated to a particle on the basis of its orientation (e.g., $a = \nwarrow$). With respect to the same contour S and the chosen particle-letter association, we now notice that a folding sequence is valid only if the contour S traverses particles in the order $a-b-c-d$ or in the reverse order $d-c-b-a$. For example, the sequence $a-b-a-d-a$ is valid, as it is a combination of the reverse and the forward traversing orders. Instead, the sequence $a-b-a-d-b$ is not valid, because b cannot follow d.

It follows that a *positional constraint* exists for the square grid, for which a curve traversing a tile (e.g., a) is constrained to proceed towards specific directions (e.g., only towards b or d, but not towards c). To better illustrate this concept, we apply a shear transformation to the triangular grid in order to transform each pair of triangles into a square, as shown Fig. 3.14. This allows us to directly compare the geometrical property of the transformed triangular grid and the square-tile grid. Starting from any tile of the square grid, movements are only allowed towards two distinct cardinal directions, as the sequence $a-b-c-d$ and its reverse need to be observed. For example, a curve traversing a tile a cannot traverse a tile of type c, but only tiles of type b or d. Conversely, a curve traversing any tile of the transformed triangular grid, can proceed traversing any other adjacent tile. This results from the fact that, regardless of the moving direction, the adjacency of particles of type A and of type V (i.e., the sequence AVAV...) is always respected.

3.4.3 Experimental Validation

To demonstrate the validity of the Force-Guiding Principle, we build the motorized prototype shown in Fig. 3.12b. A 3-particle chain hangs from a custom support with the main actuator (motor) winding up the tendons from the top (we refer to Paper $\boxed{A}$, for further details).



(a) A chain of three particles in its initial configuration.

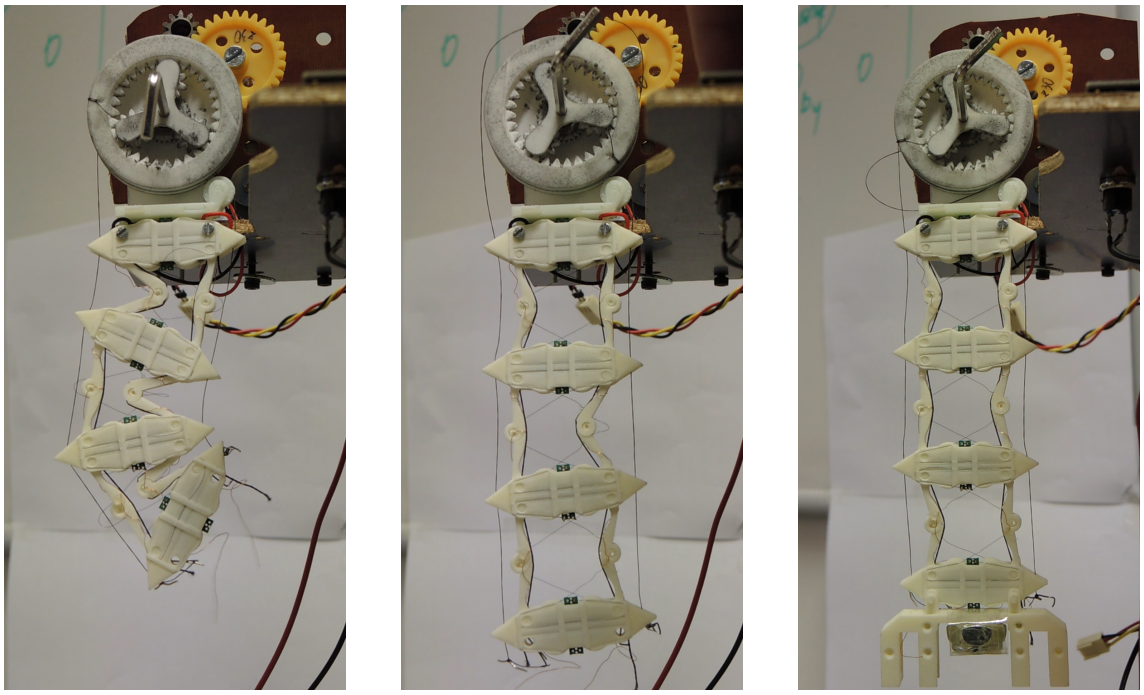
(b) Due to fabrication tolerance, a small gap exists between the edges of the second particle when fully folded.

(c) When all particles are folded, the inaccuracies due to fabrication compensate each other.

Figure 3.15: Experimental validation: the chain is folded starting from the bottom particle and proceeding upwards.

A complete actuation test, where all the particles are folded starting from the bottom and progressively proceeding upward (Fig. 3.15), demonstrates the feasibility of the triangular-particle chain as well as the validity of the Force-Guiding Principle.
The main issue we notice, while folding the chain, is a variable gap between the adjacent edges of fully folded particles (Fig. 3.15b). We ascribe this mainly to fabrication tolerance and also to the low accuracy in controlling the rotations of the actuating motor. Eventually, these fabrication inaccuracies self-compensate when all the particles are folded (Fig. 3.15c).

The transition between different shapes requires the chain to completely unfold (Fig. 3.16). This is obtained by releasing the tensile forces in order for the chain to unfold under the effect of gravity. Experiments show that the chain cannot completely unfold (Fig. 3.16b), although additional rubber bands assist the folded edges to restore the straight configuration. This is mainly a consequence of the friction at the hinges and also of the moderate weight of the components: a particle has an overall weight of about 4.5 g. An additional weight of 16 g added to the tail of the chain improves the result (Fig. 3.16c), yet without completely unfolding the chain.



(a) The chain being unfolded.

(b) The unfolding process does not completely restore the initial configuration.

(c) With the aid of an extra mass, the unfolding process slightly improves.

Figure 3.16: Experimental validation: the chain is unfolded.

3.4.4 Scalability

The static cantilever depicted in Fig. 3.17 represents the worst-case situation, where a $(N+1)$-particle chain is partially folded and only the particle head of the chain is about to fold. Assuming that the external actuator can be upgraded for larger N, the maximum force F_s the SMA wires can exert to unlock a particle entails an upper-bound to the number N of particles, for the worst-case situation.

In Paper $\boxed{A}$ we derive the following relationship between N and the actuation force F_s applied to the folding edge on the left – i.e., on the same side of the cantilever.

$$N \leq 2\sqrt{\frac{h}{2\delta} \cdot \frac{F_s}{w}} \tag{3.1}$$

where h and δ are dimensions of the folding edge, as reported in Fig. 3.17; w is the weight of a particle. In Paper $\boxed{A}$, we ignore the effect of the tendon when unlocking the left folding edge. However, the tendon can also contribute to minimize the force F'_w weighing on the folding edge and by that reduce the effort required by the force F_s. Indeed, the tension applied to the tendon can arbitrarily increase until the weight of the cantilever is completely counterbalanced, thereby nullifying the force F'_w. In this situation, an estimation of the effect of the unlocking force F'_s for $F'_w = 0$ results to be meaningless.

To correctly estimate the effects of the tensile force T produced by the tendon and balanced by the differential gear, we consider the force F_s while unlocking the right edge

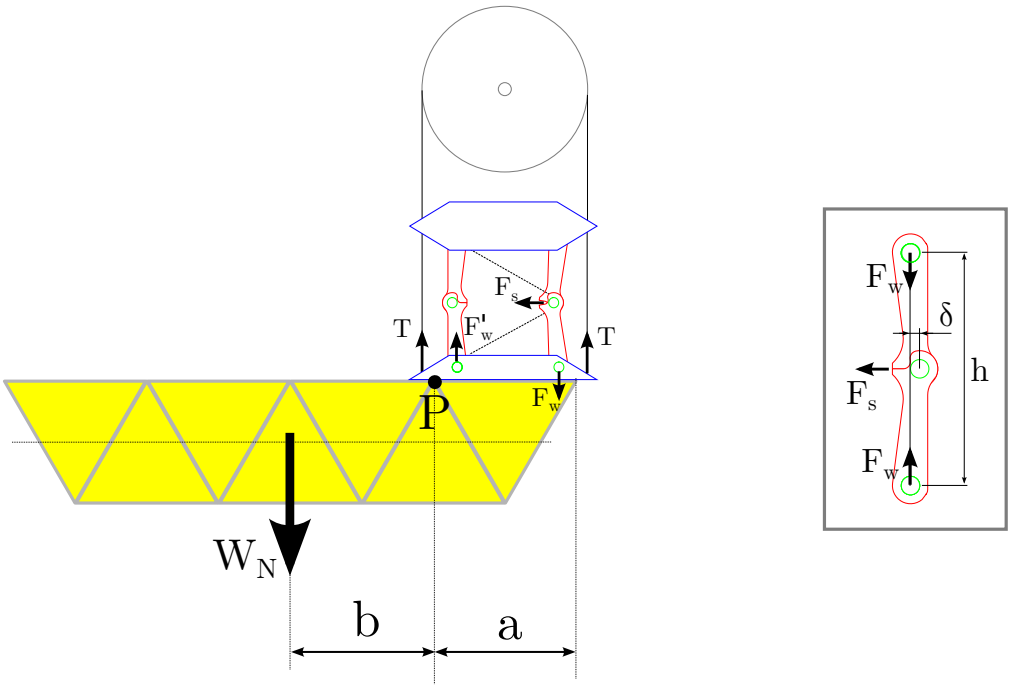


Figure 3.17: Worst case configuration of a folded chain.

(i.e., the folding edge opposite to the cantilever). By neglecting the thickness of the cantilever and the small gap between the application points of the forces T and F_w (F'_w), the resultant moment of forces about the point P and the sum of vertical forces yield the following equations:

$$\begin{cases} b \cdot W_N + a \cdot T = a \cdot F_w \\ 2\ T + F'_w = F_w + W_N \end{cases} \implies F_w = \left(2\ \frac{b}{a} + 1\right) \cdot W_N - F'_w \tag{3.2}$$

where a is the length of a particle edge; b is the horizontal distance between P and the center of mass of the cantilever; W_N is the weight of N particles; T is the tensile force exerted by the external actuator through the differential gear.

The distance between the center of mass of the cantilever and the point P, namely b, is derived as function of a and the number N of particles:

$$b = \frac{1}{4}\ N\ a - \frac{3}{4}\ a \approx \frac{1}{4}\ N\ a - \frac{1}{2}\ a \tag{3.3}$$

where, for convenience, we approximate b to simplify the following calculation. Nonetheless, we notice that through this approximation, the length of b (i.e., the length of the cantilever) is slightly overestimated. Considering that $W_N = N \cdot w$, with w the weight of a single particle, from Eq. 3.2 we obtain that:

$$F_w = \frac{N^2}{2} \cdot w - F'_w \tag{3.4}$$

As the tension on the tendon can be sufficient to nullify the load on the left edge, we assume $F'_w \geq 0$ and derive the upper bound value for N. We know that $F_s = 2\ \frac{\delta}{h}\ F_w$, as

derived in Paper [A].

$$N \leq \sqrt{2\,\frac{F_w}{w}} = \sqrt{2}\,\sqrt{\frac{h}{2\delta}\cdot\frac{F_s}{w}} \tag{3.5}$$

Compared to the result obtained in Paper [A], and reported in Eq. 3.1, the above equation shows that the maximum number of particles for the worst case situation (also taking into account the effect of the tendon) reduces by a factor $\sqrt{2}$. Nonetheless, as N is proportional to the inverse of the square root of the weight w of a particle, particles miniaturization reduces the weight w and hence increases the scalability N of the chain.

Discussion The derived relationship indicates a limit to the scalability for the worst-case situation. This leads to the concept of optimal-planning, described in the next chapter. As through the pre-planning algorithm (Sec. 3.4.2), it is possible to compute the final configuration of a chain (i.e., the folding side of its particles) before the actuation takes place, optimal planning is required to infer a chain folding sequence (i.e., which particles actuate first) to minimize the actuation forces, and hence avoid inconvenient situations, such as the one described above. The worst-case situation, for example, can be avoided or at least mitigated by prioritizing the actuation of the head particle, and then folding the remaining particles.

3.4.5 System Characterization

This section provides a characterization of our design with respect to the high-level specifications defined in Sec. 3.1.1. In particular, *resolution rendering*, *display scale*, *stability index* and *formation of 3D connected surfaces* are estimated for the prototype at hand.

Aiming at high-resolution rendering, similar consideration are provided for an SSD whose particle dimensions scale down by one and two orders of magnitude. In the second case, dimensions are comparable to those of a pixel of a 15-inch HD monitor.

As defined in Sec. 3.1.1, the number of particles (voxels) per unit volume (*ppv*) characterizes the **resolution rendering** of an SSD. This is inversely proportional to the volume V occupied by a folded particle[3], namely $ppv = V^{-1}$. As a folded particle outlines an equilateral triangle, its volume is $V = \frac{\sqrt{3}}{4}a^2 \cdot t$, with a the length of an edge and t the thickness of a chain. Considering the dimensions of a particle of the prototype at hand, *ppv* results in $ppv = V^{-1} = \frac{4}{\sqrt{3}\cdot(4.85^2)\cdot 0.8} = 0.12\,cm^{-3}$. Far from being an acceptable value, we notice that *ppv* is inversely proportional to the square of a and, assuming $a \propto t$, to the cube of a. Assuming that a and t could be scaled down by one and two orders of magnitude, Table 3.1 reports the theoretical values of *ppv* for different cases.

The maximum **display scale** is limited by the maximum length L of a chain, as the number of chains in an SSD can be in principle increased by upgrading the external actuator. Considering the maximum admissible number N of particles in a chain (Eq. 3.5), the length of a chain of *folded* particles is $L = a \cdot N/2$. With dimensions $h = 32.11\,mm$ and $\delta = 1.28\,mm$, weight[4] $w = 4.5\,g$ and force $F_s = 2.59\,kg$ (Paper [A]), it follows that for the existing prototype $N = 120$ and $L = 291\,cm$. By reducing the dimensions of a particle, the number N increases, the weight of a particle reduces, while the ratio $\frac{h}{\delta}$ and

[3]When an SSD reaches a target shape, all particles must be folded.
[4]With some abuse of notation, we express weight and forces in grams instead of Newtons.

F_s (Eq.3.5) are independent of particle dimensions. Assuming that weight and volume of a particle are proportional, Table 3.1 reports the theoretical N and L for different cases.

The **stability index** defined in Sec. 3.1.1 is the ratio $\frac{|F_{max}|}{N}$ between the maximum external force F_{max} that does not compromise the correct functioning of the system and the number N of particles in the system. The cantilever scenario shows that F_w (see Fig. 3.17) is the maximum force that a chain can withstand, without compromising its stability. Specifically, from Eq. 3.4 results that F_w is maximum when $F'_w = 0$, thus $F_w = w \cdot \frac{N^2}{2}$. As the stability index depends on the weight w of a particle, a variation of particle dimensions also affects the stability index as reported in Table 3.1. In particular, we notice that while *ppv* increases the stability index decreases.

An SSD allows the **formation of 3D connected surfaces**, and in particular of convex and concave surfaces. Perforated and disconnected surfaces can be approximated by concave surfaces with the method explained in Sec. 3.1.1. As long as every intersection between a convex or a concave surface and a plane results in a *continuous non-self-intersecting 2D curve*, a chain can outline such a curve (Sec. 3.4.2) and the SSD can display the surface. Approximation is however required in case the resulting curve is not continuous, but consists of multiple disconnected 2D curves. This requires curves to be first concatenated in a single non-self-intersecting 2D curve, by introducing additional segments and by disconnecting any possible loop. The ability of an SSD of forming 3D connected surface is independent of the particle dimensions; thus, this characteristic is not reported in the following table.

	Dimensions		– Particle –				Chain	Stability
	a	t	Volume	Weight	Number	*ppv*	Length	Index
	$[cm]$	$[cm]$	$[cm^3]$	$[g]$		$[cm^{-3}]$	$[cm]$	$[g$/particle$]$
(*)	4.85	0.80	8.15E+00	4.50E+00	120	1.23E-01	291	2.70E+02
	0.40	0.80	5.54E-02	3.06E-02	1457	1.80E+01	291	2.23E+01
	0.40	0.08	5.54E-03	3.06E-03	4607	1.80E+02	921	7.05E+00
	0.04	0.08	5.54E-05	3.06E-05	46067	1.80E+04	921	7.05E-01

Table 3.1: System characterization considering existing (*) and scaled dimensions.

A comparison between our prototype and existing architectures is reported below. The theoretical maximum length of a chain and maximum number N of particles refers to the cantilever configuration (source [114]). The stability index is derived for comparison considering the weight and the number of particles in a chain.

Robot	Mass [g]	Length [mm]	Number N	Stab. Index $[g$/part.$]$
Atron [59]	850	114	2.58	1.10E+03
Conro [17]	114	108	1.95	1.11E+02
Molecubes [125]	200	66	8.15	8.15E+02
M-Tran [75]	200	66	5.18	5.18E+02
Polybot [118]	124	58	4.19	2.60E+02
Triangular Chain	**4.5**	**2910**	**120**	**2.70E+02**

Table 3.2: Comparison between existing architectures and our prototype.

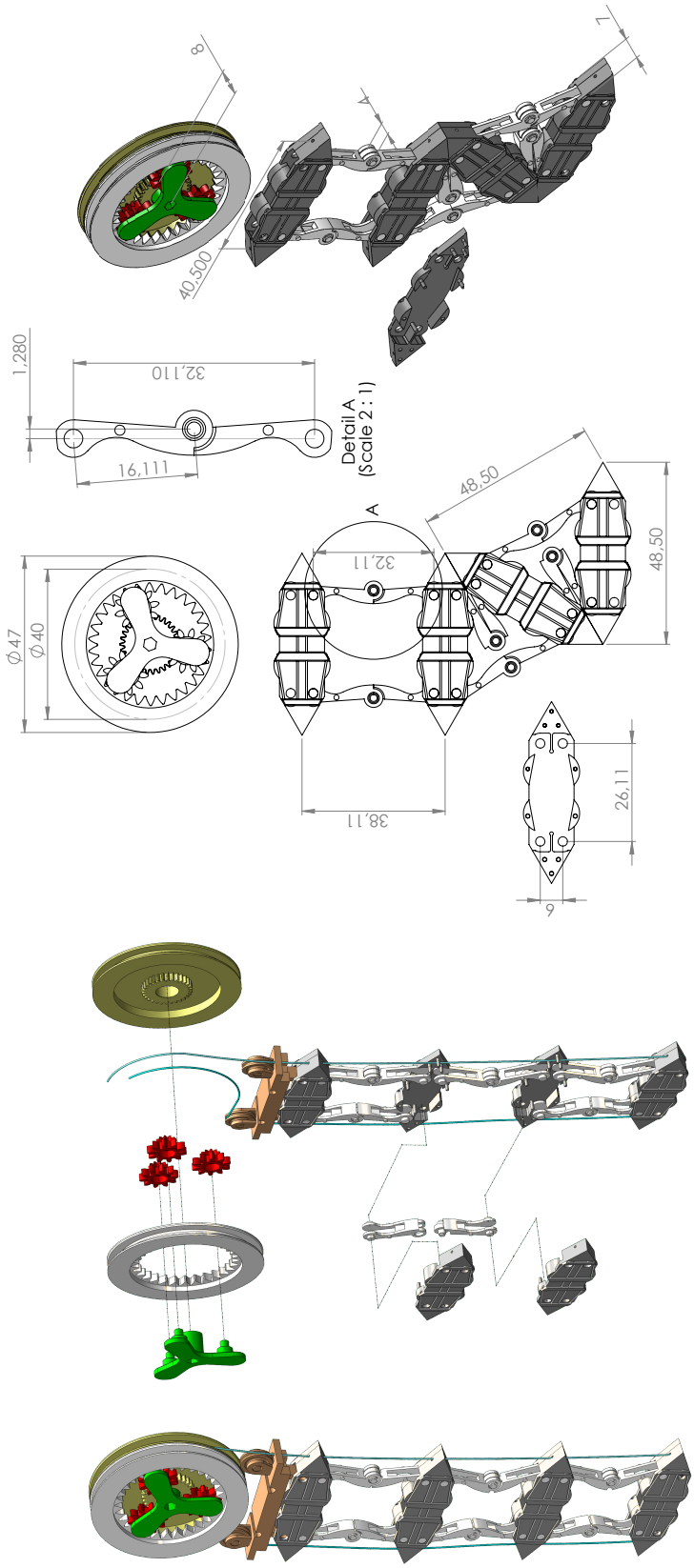


Figure 3.18: 3D model of the particle chain, exploded view, and dimensions $[mm]$.

Chapter 4

Modeling

The realization of SSD does not only raise mechatronic design challenges, it also calls for *optimal planning* and *model-predictive control* to coordinate a shape shift. This is a fundamental requirement to achieve the high *scalability* that the devised mechanical solution *potentially* allows and also to ensure mechanical stability and integrity.

In this chapter we first introduce the motivations behind optimal planning and model-predictive control, and indicate how static and dynamic models of the system are required to support planning and control by predicting the reaction of the system undergoing a shape-shift. Our contribution consists in the formulation of the static and dynamic models which should be sufficiently accurate and consistent with reality but also computationally efficient to allow planning and control in reasonable time.

4.1 Motivation and General Requirements

This section motivates the need of static and dynamic models to support optimal planning and by that to maximize system scalability.

The mechanical solution proposed in the previous chapter aims at improving scalability by means of an external actuator to remotely actuate the robotic particles. As the actuator can be arbitrarily upgraded, more particles can be added to the system. Yet scalability also depends on the shape-shifting process, as the latter influences the maximum number of particles that can be actuated. As a demonstration of this, the example reported in Sec. 3.4.4 shows that excessive actuation forces originate from inconvenient configurations, such as the cantilever, which limit the maximum number of particles in a chain. This mainly derives from leverage effects that significantly increase the actuation forces beyond the capability of the system and even above the maximum forces necessary to eventually retain the final shape. To avoid these situations, static and dynamic models of the system are required to predict the behavior of Shape-Shifting Display (SSD) and enable planning and control algorithms to find optimal solutions.

Optimal planning. A sequence of *intermediate configurations* is required to progressively actuate the system. Indeed, it is not possible to actuate all particles simultaneously because of the existing electromechanical constraints (a particle draws 300 mA to unlock a folding side). Each *intermediate configuration* selects a subset of unfolded particles in each chain to be actuated. This calls for *optimal planning* techniques to identify a sequence

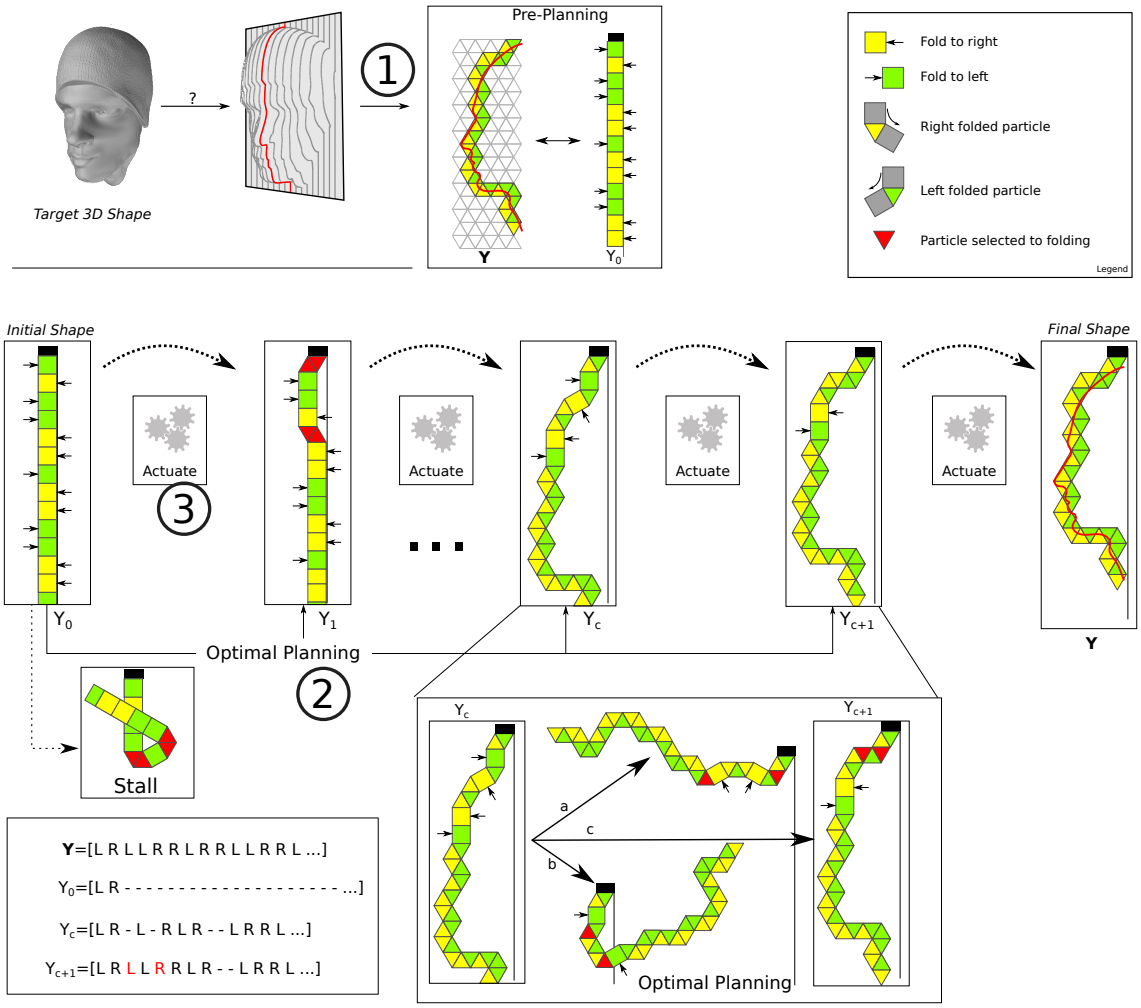


Figure 4.1: Shape-Shifting Process: intermediate configurations are required to minimize the actuation forces.

of intermediate configurations that minimizes, or at least limits, the maximum actuation forces. To predict the forces acting on the system during its actuation, static and dynamic models are derived to support optimal planning.

Model-predictive control. To minimize the actuation forces a model-predictive control, compensate for the absence of sensors (Sec. 3.4), which are not included in the minimalistic design of a particle in order to contain its costs and complexity. This means that the folding state of chains composing an SSD is not measurable and its not possible to enable any feedback loop. To compensate for that and properly actuate the system, model-predictive control estimates through kinematic and dynamic models of the system the minimal actuation forces necessary to obtain a shape shift.

4.1.1 Shape-Shifting Process

Fig. 4.1 depicts the shape-shifting process that essentially consists of three phases:

1. For each chain in the SSD, compute the *final* folding state (i.e., left/right folding of each particle).

2. For each chain, compute the sequence of intermediate configurations, in which only a subset of particles folds (an equivalent number of particles must fold in every other chain of the SSD).

3. Progressively actuate the system, according to the planned solution.

Given a target 3D shape, the pre-planning algorithm (Sec. 3.4.2) determines for each chain of the SSD the target folding configuration to approximate a corresponding slice of the target 3D shape. A folding configuration (or chain configuration) is a binary sequence that indicates the folding side of each particle in the chain. For example, to approximate the head profile highlighted in Fig. 4.1, the folding configuration $\mathbf{Y}$ depicted in the "Pre-Planning" frame indicates the target folding side of each particle. As long as the contour to be approximated is a planar non-self-intersecting curve, the computation of a chain configuration is possible (Sec. 3.4.2).

As it is not possible to actuate all the particles in the system simultaneously, starting from the initial configuration Y_0 and aiming at the target configuration $\mathbf{Y}$, a sequence of intermediate configurations $[Y_1 \cdots Y_c \cdots]$ needs to be identified to progressively fold the chains of the SSD. At each intermediate configuration a *subset* of particles in each chain is selected to actuate. Even though the final configuration $\mathbf{Y}$ complies with the physical constraints imposed by the mechanical design, the selections of particles to fold is critical, because it might lead to mechanically infeasible situations. We identify two critical cases: "stall" and "overload".

"Stall" occurs when the selection of the particles to fold cause a chain to self-intersect. An example is shown in Fig. 4.1, where from the initial configuration Y_0 four right-folding particles are selected to actuate. This causes the chain to "overlap" itself, which is not physically admissible. A proper actuation should instead select a different subset of four particles as shown in configuration Y_1. Optimal planning is required to identify proper intermediate configurations to avoid any stall condition.

"Overload" occurs when a folding operation provokes too intense actuation forces. If not properly planned, the sequence of intermediate configurations might cause actuation forces to be even stronger than the forces necessary to eventually retain the target shape $\mathbf{Y}$. This reduces the stability of the system and might also permanently compromise its integrity. An example is given in Fig. 4.1. Let us assume that the transition between the intermediate configurations Y_c and Y_{c+1} requires only two particles to be folded out of the four still unfolded (arrows and colors indicate the target folding sides). Among the possible folding selections, three cases "a", "b" and "c" are highlighted in lower frame of Fig. 4.1. As Fig. 4.1 shows, selection "c" is the one that minimizes the required actuation forces. The alternatives "a" and "b" cause overload because a long tail of the chain must be lifted. This would cause intense actuation forces to reduce the scalability of the system, as reported in Sec. 3.4.4. Indeed, the cantilever configuration described in Sec. 3.4.4 is the worst case overload scenario, which limits the scalability of the system to a maximum number N of particles proportional to the square root of the force F_s, i.e., $N \propto \sqrt{F_s}$; F_s is the force that the weak actuator embedded in each particle can exert to unlock a folding side of the particle.

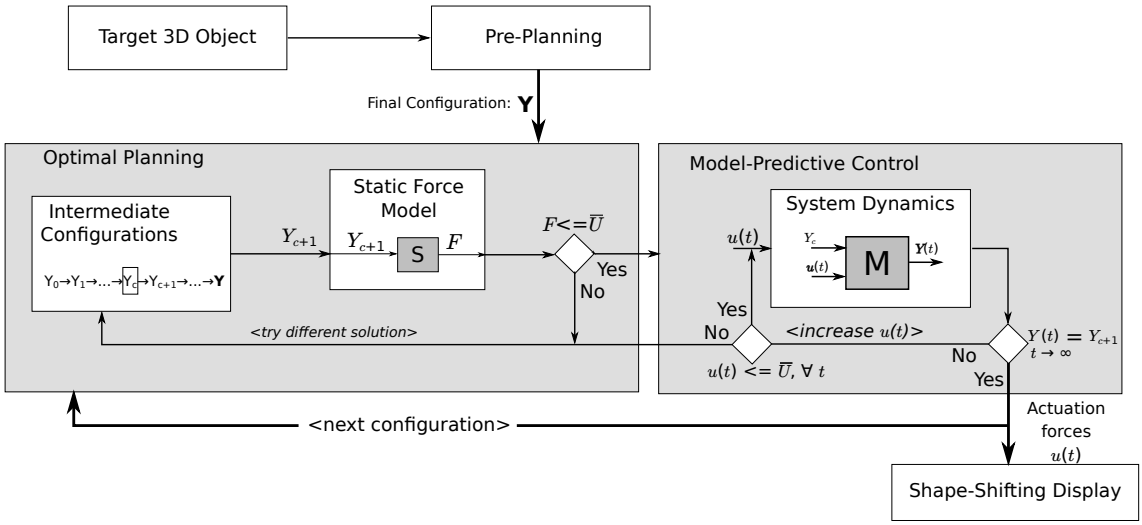


Figure 4.2: Optimal planning and control, overview.

Optimal planning is required to identify a sequence of intermediate configurations where actuation forces are minimized, or at least limited within an acceptable range. Model-predictive control supervises the transition between consecutive intermediate configurations and estimates the minimal forces necessary to properly actuate the system. This approach is required to compensate for the absence of sensors embedded in particles, which does not allow any feedback control strategy.

4.1.2 Optimal Planning and Model-Predictive Control

The flow diagram of Fig. 4.2 depicts the complete shape shifting process including actuation. The pre-planning generates the final configuration $\mathbf{Y}$ for each chain of the SSD. Optimal planning seeks an admissible sequence of intermediate configurations, aiming to avoid *stall* and *overload*. Model-predictive control is applied to infer a set of minimal actuation forces to execute the transition between consecutive configurations.

The selection of an optimal sequence of intermediate configurations is a fundamental requirement for the system, to ensure its stability and integrity as well as to increase the overall scalability. The optimization problem does not only involve a single chain, but involves all chains as they depend on the same single external actuator. Therefore, an equivalent number of particles needs to be folded in each chain at each reconfiguration step. In the following, we assume this number of particles to be known in advance.

As the planning of a shape shift requires a large set of possible intermediate configurations to be explored, we split the planning process into two phases. The first phase consists in excluding possible intermediate configurations, which certainly lead to too intense actuation forces. This is done through the application of a *static model* that predicts the static forces acting in the system at equilibrium. The static model has the advantage of being simple and computationally efficient, although not very accurate. It is nonetheless a reasonable trade-off, in order to reduce the set of possible configurations. The second phase adopts a more accurate dynamic model of the system, to asses the feasibility of the

found shape-shift solution.

A static-force model is derived to provide efficient yet acceptable predictions for excluding inconvenient intermediate configurations. As Fig. 4.2 indicates, optimal planning seeks a sequence of intermediate configurations $[Y_0 \cdots Y_c \cdots]$ whose combination results in $\mathbf{Y}$. A configuration (e.g., Y_{c+1}) indicates a subset of particles to be folded. Each transition Y_c to Y_{c+1} should not exceed the maximum actuation forces admissible in the system. To efficiently exclude inconvenient solutions a static model is applied to predict the forces F necessary to retain Y_{c+1} at equilibrium. In case such forces are above the maximum admissible limit $F > \overline{U}$ a different subset of particle is chosen to be folded.

The static-force model does not provide a comprehensive prediction of the actuated system. To verify the feasibility of the found solutions and actuate the system subsequently, a model-predictive control approach is devised. An accurate model of the system is derived to predict the dynamics of the whole SSD upon the application of control input $u(t)$, which is the set of actuation forces within the range $\overline{U}$. The dynamic model can infer, starting from the current configuration Y_c, the behavior $Y(t)$ of the system over a certain time horizon. If the actuation of the system through the application of control input $u(t)$ eventually leads to the configuration Y_{c+1}, such a configuration is feasible and the control input $u(t)$ is applied to the SSD for actuation. Instead, if the intensity of the control input $u(t)$ is not sufficient to reach the configuration Y_{c+1}, heuristic techniques are applied to adjust the input $u(t)$ to properly actuate the system. If it is still not possible to find a configuration $u(t) \leq \overline{U}$ such that the system eventually reaches the target configuration $Y(t) = Y_{c+1}$, the intermediate configuration is said to be not feasible, and another solution needs to be sought in the state space.

Heuristic techniques are used in combination with the forward dynamic model to estimate a set of minimal forces to actuate the system. An appropriate approach would be to derive such forces through application of an inverse dynamic model. However, the derivation of an inverse dynamic model is particularly difficult for an underactuated system (each particle in the chain has three Degrees of Freedom (DOFs) and only two control inputs, which result in an underactuated system) especially when compliant elements, such as the tendons, are present [8].

4.2 Static Model

The static model provides a simple and computationally efficient solution to compute the static forces acting on a chain at equilibrium. In this section, we define the requirements and the modeling methodology to guide the subsequent formulation. The formulation takes into account the tension that propagates along a foldable chain and the forces acting on the chain at equilibrium. A validation of the model demonstrates its consistency with reality. A final discussion about the intrinsic limitations of the static model motivates the requirements for a more accurate dynamic model, described in the following section.

4.2.1 Requirements

The following requirements are considered for the static model:

1. The static model predicts the tensile forces acting in a chain at equilibrium. Such forces are necessary to retain the target shape and any intermediate configurations.

A configuration (final or intermediate) is said to be feasible when the physical constraints allow the chain to take on such a configuration and the forces acting in the chain to retain such a configuration (system at equilibrium) are within acceptable limits ($F \leq \overline{U}$, as in Fig. 4.2).

2. The model must applicable to a chain having particles in each possible folding state: unfolded, partially folded, or completely folded.

3. As the static model is applied during the initial planning phase to explore the state space of the system and exclude infeasible configurations, the implementation of the derived model should be computationally efficient, given the large dimension of the state space.

Input and Output. The block "S" in Fig. 4.2 represents the static model with its input and output. Input of the model is the target configuration of the chain (e.g., Y_{c+1}), for which the model predicts the tensile forces necessary to retain the corresponding shape, and the actuation forces to *slowly* fold particles (i.e., quasi-static condition). Output of the model is the predicted actuation force (e.g., F) that the external actuator needs to exert to actuate a chain.

4.2.2 Related Work

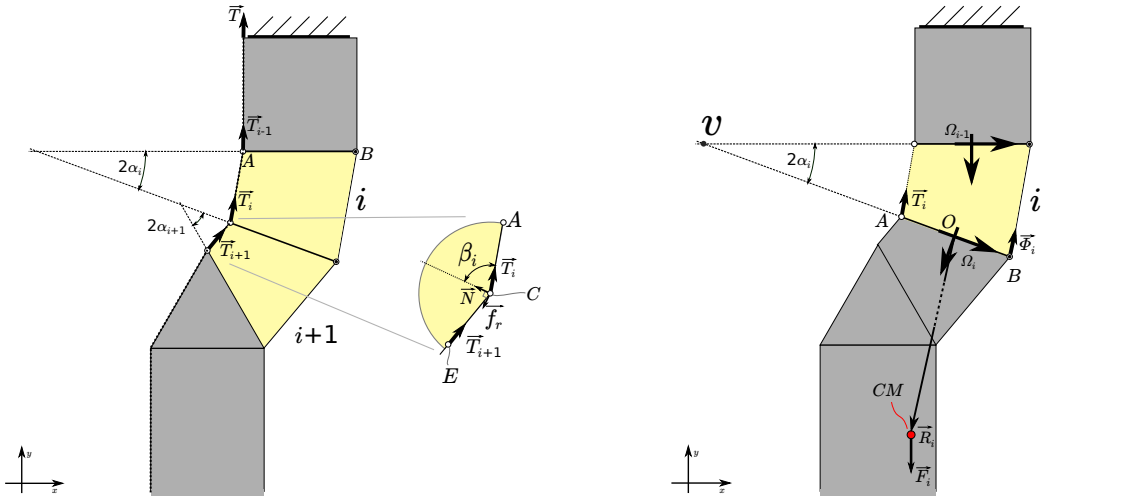
A foldable chain is an underactuated tendon-driven robotic system. Tendon-driven robotics allows the creation of lightweight structures [31,62,70], suitable for the realization of tiny manipulators with reduced inertia of both the robotic arm and the end-effector [3]. As in general the most relevant aspect in such systems is the positioning of the end-effector, rather than the posture of the whole robotic arm, physical models presented in the referenced works are hardly adaptable to predict forces acting on our system.

As we aim at modeling the tensile forces needed to retain a folding configuration *at equilibrium*, we assume all particles to be "frozen" at each observation time, namely they are in a condition of equilibrium. In this way, the static model only need to encompass forces acting on interconnected rigid bodies [72] for which the compliance of the tendon is not relevant.

Tendon-driven continuum robotics [15, 31, 62] is more closely related to a foldable particle chain. The main difference is that the shape that such robots can eventually outline is not piecewise controllable, because the number of controllable Degrees of Freedom (DOFs) is limited to the number of tendons in use. In our system, by introducing local actuators in each particle (Force-Guiding Principle in Sec.3.2.3 and Sec. 3.4.1), the number of controllable DOFs increases, event tough the system results underactuated. Nonetheless, the cited works suggest an approach to model the tension propagating through the system that can be adapted to model our chain.

4.2.3 Modeling Methodology and Assumptions

To effectively and efficiently model the forces acting in a Shape-Shifting Display (SSD), we focus on one single chain. This does not limit the generality of the derived model, considering that the differential gears integrated in an SSD decouple the actuation forces among different chains, which are independent of each other. Nevertheless, the effort of



(a) Friction propagation model. **(b)** Forces acting on the chain at equilibrium.

Figure 4.3: Chain at static equilibrium.

the external actuator corresponds to the sum of the forces necessary to actuate each single chain.

In order to obtain a simple and computationally efficient model (Req. 3), the following assumptions facilitate the subsequent model formulation:

1. Particles in an N-particle chain are uniquely identified by increasing numbers, having head and tail particles corresponding to 1 and N.

2. The top and bottom edges of a particle outline a rectangle when unfolded.

3. The top and bottom edges of a particle outline an equilateral triangle when fully folded.

4. During the transition between unfolded and folded state, i.e., when the particle actuates, the rectangular shape gradually transforms, first into an isosceles trapezoid whose short base corresponds to the folding side of the particle, and eventually into an equilateral triangle.

5. A local frame of reference is attached to the top edge of each particle. The frame of reference of the head particle is an inertial frame of reference.

6. Tendons pass through eyelets, which are considered dimensionless points.

7. When observing the effect of forces on a folding particle, we assume the rest of the chain to be "frozen", as all particles are at equilibrium.

8. Friction at the hinges is negligible, since a quasi-static model is applied.

9. The internal friction of a differential-gear is negligible, thus the tensions applied at the two extremes of the tendons actuating a chain are equivalent.

10. We neglect the small gap between eyelets and hinges. This is equivalent to neglect the small leverage effects due to a tendon compressing a non-folding side.

Fig. 4.3b illustrates the first seven assumptions listed above. In particular, all particles except for particle i are in the frozen state: particles with lower index do not move and thus are ignored; the frozen tail, composed of particles in range $[i+1, N]$, is physically represented by a mass $M_i = m \cdot (N - i)$ concentrated in the center of mass of the tail, where m is the mass of a single particle. This mass, subject to gravitational acceleration, is responsible for the force F_i applied to the tail of the chain.

4.2.4 Model Formulation

We aim at modeling the static tensile force T necessary to actuate an N-particle chain, as depicted in Fig. 4.3b. Assumptions 9 and 10 allow us to only focus on one tendon acting on the chain, as similar considerations apply to the other one.

To efficiently formulate a static force model, we address the following two problems separately: 1. How do the tension forces propagate along the chain? 2. What is the intensity of the forces acting on a chain at equilibrium?

Tension Propagation Model. Referring to Fig. 4.3a, the tensile force T results from the contribution of the forces acting on each particle. A loss of tension is considered for a tendon traversing an eyelet, due to the friction force F_r counteracting the action of the tendon [31,62]. Consequently, the tension T_i of the tendon entering eyelet i propagates to the next eyelet $i+1$ with an intensity $T_{i+1} = T_i - F_r$. Assuming a dimensionless eyelet (assumption 6), all the forces concentrate in one single point.

The well-known Amonton-Coulomb friction model [89] is applied to estimate the friction force $\overrightarrow{F_r}$, proportional to the normal force $\overrightarrow{N}$ by the coefficient of static friction μ, i.e., $F_r = \mu \cdot N$. The latter depends on the material of both the tendon and the particle eyelet. The force $\overrightarrow{N}$ is the force that the tendon exerts on the eyelet C along the line bisecting the angle $\widehat{ACE}$ (Fig. 4.3a). The angle β_i enclosed by the vectors $\overrightarrow{T_i}$ and $\overrightarrow{N}$, as indicated in Fig. 4.3a, is half the deflection angle of the tendon passing through the eyelet. The latter is a function of the folding state of the two adjoining particles $i-1$ and i, specifically $\beta_i = \frac{\pi - \alpha_i - \alpha_{i+1}}{2}$. Considering that the intensity of the normal force $N = T_i \cdot |\cos\beta_i|$, hence the friction force $F_r = \mu \cdot T_i \cdot |\cos\beta_i|$, the tension that propagates to the next particle is:

$$T_{i+1} = T_i - F_r = T_i\big(1 - \mu \cdot |cos(\beta_i)|\big) = T_i\left(1 - \mu \cdot \left|\sin\left(\frac{\alpha_i + \alpha_{i-1}}{2}\right)\right|\right) \tag{4.1}$$

The above equation can be recursively applied to compute the tensile force T_N starting from the force T, namely the actuation force that the external actuator exerts on the tendon. We consider that $T \equiv T_0$ and assume that the segment of tendon interconnecting the external actuator and the chain is perpendicular to the top edge of the head particle, hence $\alpha_0 \equiv 0$. As the opposite extreme of the tendon is tied to the tail particle N, the tension T_N does not propagate through the last eyelet, but applies entirely to the eyelet itself.

Force Model. In order to model the actuation forces, we analyze the forces acting on an N-particle chain at equilibrium state. As depicted in Fig. 4.3b, the "frozen" assumption

(7) facilitates this task. The chain reduces to a mechanical system composed of two rigid bodies (the frozen particles $[1, i-1]$ and $[i+1, N]$) interconnected by a "deformable" body (the folding particle i) subject to the tensile force T_i.

As the particle head of the chain is steady according to assumption 5, all the frozen particles $[1, i-1]$ are steady too. Consequently, the reference frame Ω_{i-1} attached to the top edge of particle i is an inertial frame. The tensile force T_i applied to the bottom eyelet of particle i is the force necessary to lift the frozen tail of the chain, as no tension propagates through the frozen tail. As Fig. 4.3b illustrates, at equilibrium, there are three forces acting in the system and generating moments, which are the tensile force T_i applied in A, the weight of the frozen tail F_i applied at the center of mass "CM", and the constraint force ϕ_i applied at the joint B. Due to assumption 4, the torques at the hinge elide each other. Let R_i be the displacement of CM with respect to the frame Ω_i, the following static equations must hold at equilibrium:

$$\overrightarrow{R_i} \times \overrightarrow{F_i} + \overrightarrow{OA} \times \overrightarrow{T_i} + \overrightarrow{OB} \times \overrightarrow{\phi_i} = \overrightarrow{0} \tag{4.2}$$

$$\overrightarrow{F_i} + \overrightarrow{T_i} + \overrightarrow{\Phi_i} = \overrightarrow{0} \tag{4.3}$$

where the resultants of the moments (4.2) and of the forces (4.3) are null. The combination of the two equations, where ϕ_i is replaced by $\phi_i = -\overrightarrow{F_i} - \overrightarrow{T_i}$ according to Eq. 4.3, yields the following equation solvable for $\overrightarrow{T_i}$.

$$\overrightarrow{AB} \times \overrightarrow{T_i} = (\overrightarrow{R_i} - \overrightarrow{OB}) \times \overrightarrow{F_i} \tag{4.4}$$

which is the equation of a lever having fulcrum in B. In the above equation $\overrightarrow{AB}$ is a structural parameter of a particle – i.e., the length of an edge, from which $\overrightarrow{OB}$ can be derived as O is the mid-point of $\overrightarrow{AB}$. The unknown force F_i and the position of the center of mass R_i needs to be derived.

Let $2 \cdot \alpha_i$ be the angle enclosed by two consecutive reference frames Ω_{i-1} and Ω_i, The force F_i is computed with respect to the reference frame Ω_i as:

$$\overrightarrow{F_i} = (N - i) \cdot m \cdot \overrightarrow{g_{[\Omega_i]}} \tag{4.5}$$

where m is the mass of a particle and $\overrightarrow{g_{[\Omega_i]}}$ the gravitational acceleration referenced in Ω_i, i.e., it is the vector $\overrightarrow{g}$, defined in absolute coordinate system (x, y), and rotated by the angle $(\pi - 2\sum_{t=0}^{i-1} \alpha_t)$.

As a folding particle outlines a regular geometry (trapezoid) according to assumption 4, it is possible to define a function $X(\alpha_i)$ indicating the position of the center of mass of particle i, with respect to the local reference frame Ω_{i-1} and α_i. Let $M(\alpha_i)$ be a transformation between the frames Ω_i and Ω_{i-1}, such that $P_{[\Omega_{i-1}]} = M(\alpha_i) P_{[\Omega_i]}$, where $P_{[*]}$ is a point referenced in the indicated frame. The position R_i of the center of mass "CM" of the frozen tail, which includes particles $[i+1, N]$ derives as:

$$\overrightarrow{R_i} = \frac{1}{N - i} \sum_{t=i}^{N} M_{\Omega_t}^{\Omega_i} \overrightarrow{X(\alpha_t)} \tag{4.6}$$

The tension T_i can thus be estimated from the above equation and equations 4.4 and 4.5

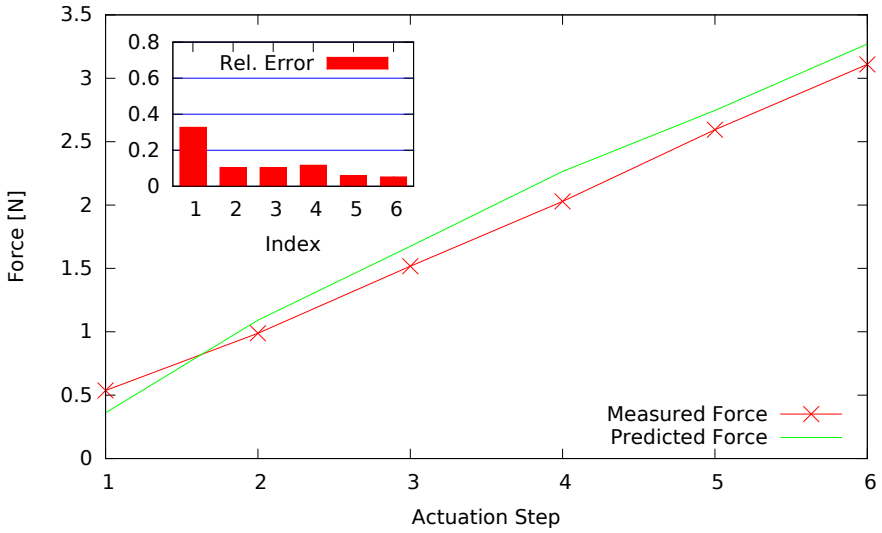


Figure 4.4: Forces acting on the chain at equilibrium.

4.2.5 Validation

By means of a manually actuated two-particle-chain prototype, a predecessor of the prototype of Fig. 3.12, we create an experimental setup to reproduce the situation shown in Fig. 4.3b. In order to simulate the effect of a long tail, we extend the two-particle chain with a rectangular plastic sheet, of known mass and dimensions. We fix the head of the prototype to a horizontal bracket and measure the tensile force necessary to fold the second particle of the chain. If we assume that the second particle of the chain under observation, corresponds to the particle i of Fig. 4.3b, the force that the we measure is the tensile force T. The extra mass attached to the tail of the chain, has its center of mass at the position R_i with respect to the top edge of particle i, as indicated in Fig. 4.3b. As we know the dimensions and geometry of the extra mass as well as its relative position with respect to the bottom edge of particle i, we can calculate R_i.

By means of a spring employed as dynamometer, we apply and measure the tensile force T exerted on the tendon when folding the chain. Simultaneously, we measure the linear translation of the tendon pulled off the chain in order to infer the folding angle α of the particle under observation.

The validation of the above formulated model is obtained through comparison between the predicted force T' and the experimental force T, at different actuation steps. The graph of Fig. 4.4 reports the results. We notice that, after an initial divergence, the predicted and experimental forces follow a similar trend, with a relative error lower than 0.12. Our conjecture is that the initial divergence between the two setups derives from mechanical backlash at the joints of the prototype. As this causes a non-linearity in the real system, the measurement of the α value at the first actuation step is not reliable and leads to misleading observations.

4.2.6 Limitations

The derived static model presents some limitations we need to consider.

1. A limitation of the static model is that it does not take into account situation when multiple partially folded particles tend to arrange into a state of minimal energy sliding along the tendon.

2. Also, by neglecting friction at the hinges (assumption 8), forces to actuate the chain are underestimated, while forces to retain a shape are overestimated.

3. It is not possible to estimate the mechanical stress on the components of the chain.

4. By assuming that particles fold regularly, we ignore possible shear forces between the top and bottom edges of a particle, which could accidentally unlock the lateral side of the particle.

5. The model does not indicate whether the force to unlock the folding side of a particle is sufficient, for which the equation (Eq. 3.5) reported in Sec. 3.4.4 applies.

6. The model applies to a single chain and does not take into account the effect of multiple chains actuated simultaneously.

The dynamic model presented in the next section addresses and overcomes the following limitations: Lim. 1, Lim. 2, Lim. 4, Lim. 5, Lim. 6.

4.3 Dynamic Model

The static model derived in the previous section predicts whether the maximum actuation forces are sufficient to *retain* an intermediate configuration. While this condition is *necessary* for assessing the feasibility of a target shape, it is not *sufficient* to ensure that also the actuation forces to execute the transition between *consecutive intermediate* configurations do not exceed the maximum limit. Too intense actuation forces might indeed occur in case of incorrectly planned intermediate configurations, which undermine the stability and the integrity of the whole system. Considering that the effect of the inertial forces and moments becomes dominant as the length of the chain and the speed of actuation increase, the static model previously proposed is not sufficient to accurately predict the behavior of the system during the transition between consecutive intermediate configurations.

A dynamic model is therefore derived in this section to overcome this and other limitations inherent to the static model. However, considering that the static model is computationally more efficient than the dynamic model, the two models are complementarily applied: through the static model we preselect a set of potentially feasible intermediate configurations, by excluding certainly infeasible configurations (e.g., those calling for too intense retain forces); the dynamic model is subsequently applied to verify whether the preselected configurations are actually feasible and thus the actuation forces to execute a shape-shift do not exceed the maximum limit. This approach takes advantage of the reduced computational complexity of the static model, which makes the exploration of the state space more efficient. At the same time, the correct actuation of the system relies on the dynamic model, which despite being less computationally efficient than the static

model, is more accurate and reliable than the latter. In addition, the application of the dynamic model allows us to infer a set of minimal actuation forces necessary to execute the transition between consecutive intermediate configurations, in this way enhancing the scalability of the system (see Sec. 4.1.2).

In the following, we first define a set of requirements to drive the formulation of the dynamic model. As our aim is to model the dynamics of the whole Shape-Shifting Display (SSD), we adopt a modular approach that allows us to analyze each component separately and eventually combine the corresponding sub-models. The mechanical coupling between components indicates how sub-models need to be combined. As for some components existing dynamic models can be used (e.g., for the motor), we focus on a single chain for which we derive kinematic and dynamic equations. We propose two different formulation approaches: chain-oriented, which defines the dynamics of a chain as a whole; and particle-oriented, which focuses on the dynamics of the elementary particle and thereby infers the dynamics of the whole chain. An evaluation and comparison of the two formulations is provided as conclusion of this chapter.

4.3.1 Requirements

The dynamic model shall overcome the following limitations (Lim.) of the static model (Sec. 4.2.6): Lim. 1, Lim. 2, Lim. 4, Lim. 5, Lim. 6. In addition, we consider the following requirements.

Modularity. The hierarchical architecture of an SSD is suitable for a modular modeling approach, whereby each sub-component is analyzed and modeled independently and eventually the corresponding sub-models are integrated. This allows us to use existing models of the most common components (e.g., motor). The integration of the sub-models into a single model predicting the behavior of the whole system can be done by taking into account the mechanical coupling among components, namely their reciprocal influence.

Completeness. Eventually an SSD should be modeled in all its parts: the external actuator, the differential winches, the chains, and the particles. This overcomes Lim. 6 of the static model. In addition, we aim at modeling a particle considering the dynamics of its links, such that the effect of shear forces applied to top and bottom edges are also taken into account, overcoming Lim. 4. This is necessary to verify, for example, whether a particle accidentally unlocks under the effect of externally applied forces.

Controllability. A chain is an underactuated system, as one control input (i.e., the compressing force due to the external actuator) is responsible for a larger number of Degrees of Freedom (DOFs) (i.e., an unlocked particle has two DOFs). To ensure correct actuation of the system, the tendons exerting the compressing force must be tight at any time, in order for the system to remain controllable. The dynamic model should indicate whether this condition is met.

Input and Output. As the block "M" in the model-predictive control of Fig. 4.2 (on page 66) indicates, inputs to the model "M" are the current configuration Y_c and a set of actuation forces $u(t)$, while the output $Y(t)$ is the behavior of the system during a time window corresponding to the prediction horizon (i.e., the time to transition between consecutive intermediate configurations).

4.3.2 Related Work and Challenges

The modular structure of an SSD allows us to model each component independently and eventually integrate the corresponding sub-models. To define the interaction among components, existing techniques like "Power-Oriented Graph" [123] take into account the "power transfer" between mechanically coupled components, in terms of action-reaction pairs. For example, a motor induces an angular velocity ω on a drive shaft (action), while the latter – e.g., due to inertial effects – counteracts with a torque τ that applies to the motor (reaction). The product of angular velocity and torque $\omega \cdot \tau$ corresponds to the amount of power being transferred between the two systems [123]. By systematically applying this approach to each pair of coupled components, we can identify input and output of each sub-model in terms of the action-reaction pairs. An advantage of this approach is that existing dynamic models of the motor and differential gears can be used and integrated. For this reason, in the following we focus on the dynamics of a chain.

A chain is an *underactuated* planar multibody system, which consists of rigid links interconnected via revolute joints. It is actuated by a pair of tendons exerting a compressing force on each modular particle composing the chain. When a particle of the chain is unlocked to fold on one side, only one input (the compressing force due to the external actuator) controls the two DOFs of the particle. In an N-particle chain, when M particles are unlocked ($M \leq N$) the compressing force is the only input to control the $2 \times M$ DOFs of the chain, which makes the chain an *underactuated system.*
As the ultimate goal of the dynamic model is to support optimal planning and control algorithms to predict the behavior of the chain and to infer a set of minimal actuation forces, an inverse dynamic model should be applied to compute the forces necessary to achieve a target behavior. However, as the derivation of such an inverse dynamic model is particularly difficult for the system at hand because of the compliant elements [8] (i.e., the tendons) and the fact that a chain is an underactuated system, we derive a forward dynamic model and adopt the heuristic methods described in Sec. 4.1.1 to infer minimal actuation forces. Nevertheless, the derivation of a forward dynamic model remains quite challenging because of the presence of compliant elements and the mechanical loops that the six links of a particle form.

To formulate kinematic and dynamic equations, a combination of absolute coordinates and minimal coordinates [49] is generally used. However, the constraints due to closed kinematic loops (e.g., the six links of a particle forming a loop) do not allow us to formulate equations of motion in minimal form [102]. Traditional approaches face this problem by temporarily "cutting" the closed loop [18] and by formulating equations of motion of the resulting "tree-type multibody systems" [32, 33, 35, 37, 100, 103]. Subsequently, to restore the loop closure [18], an equivalent set of constraint forces is derived in form of Langrange's multiplier [66] and introduced in the dynamic model.
A major limitation of the methods listed above is that they can be applied to systems consisting of only rigid bodies, while they do not address the case of rigid and compliant elements combined together. As in a particle chain the compliance of the tendons is also relevant, additional insights and considerations are required in order to derive a valid dynamic model and overcome the limitation found in [32, 33, 35, 37, 100, 103].

Lee et al. propose in [68] a modeling methodology specific for tendon-driven robots. Their target system is a robotic arm where a set of pulleys guide the tendons along the arm.

Each revolute joint is remotely actuated by means of a dedicated pair of counteracting tendons, which control direction and velocity of rotation. Our system instead consists of only two tendons driving all the particles in the chain regardless of the number of revolute joints. This is mainly due to the fact that a dedicated pair of tendons for each particle would limit the scalability of the system and increase its costs. Thus, as the modeling methodology proposed in [68] is not suitable for being applied to our system, a modeling challenge exists.

4.3.3 Modeling Methodology

We consider the hierarchical architecture of an SSD and identify four principal components: 1. motor and drive shaft, 2. differential gears and winches, 3. chain, and 4. particle. To derive the dynamic model of the whole system, we follow a modular approach, whereby the four components are first modeled independently and eventually the resulting sub-models are integrated. This approach has two advantages: first, we can use existing models of the more common components (e.g., motor [123]); second, we can focus on the model of a particle chain which is the most critical component of the system.

Modular Modeling. To effectively model each component and to eventually be able to combine the sub-models, we need to understand how components are mechanically coupled, namely how they influence each other. To this end, we adopt the concept of "Power-Oriented Graph" [123, 124] and describe the mutual action-reaction of coupled components in terms of the "physical quantities" (e.g., torque and angular velocity) responsible for the power being transferred between these components. The diagram in Fig. 4.5 depicts this concept showing the four components and the action-reaction pairs. For example, the motor and the drive shaft induce an angular velocity ω (action) on each differential winch, which reacts with a torque τ counteracting the rotation of the motor (reaction). Similarly, the rotation of the differential winch provokes a linear motion $\dot{r}$ of the tendons folding a chain, which react with a tensile force T counteracting the induced motion. Through the differential winches the tensile force T is converted into the torque τ, and since many chains are connected through the same shaft to the motor, the overall reaction forces that the motor experiences, is the sum of the reaction torques due to each single chain. The products $\omega \cdot \tau$ and $T \cdot \dot{r}$ correspond to the power being transferred between the motor and the differential winches. If we assume no friction at the differential winch, then $\omega \cdot \tau = T \cdot \dot{r}$ holds as the energy is entirely transferred through the differential winches, without dissipation.

The motion $\dot{r}$ of the tendon being pulled up, causes a motion $\dot{r}_i$ of the bottom edge of particle i, which is due to the interaction between tendons and eyelets. As the edges of consecutive particles coincide, also their local frames of reference (e.g., Ω_i for particle i) must coincide. Consequently, the reaction forces F_i associated to the bottom edge of particle i must equal the reaction force F_{i+1} of the top edge of the next particle.
The tensile force T results from the propagation of the reaction forces T_i that depends on the friction between tendons and eyelets, on the force F_i and on the gravitational force.

Particle Chain. As the particle chain is a key element for the system and can be modeled independently of the other components, we focus on the chain to analyze its dynamic response when actuated.

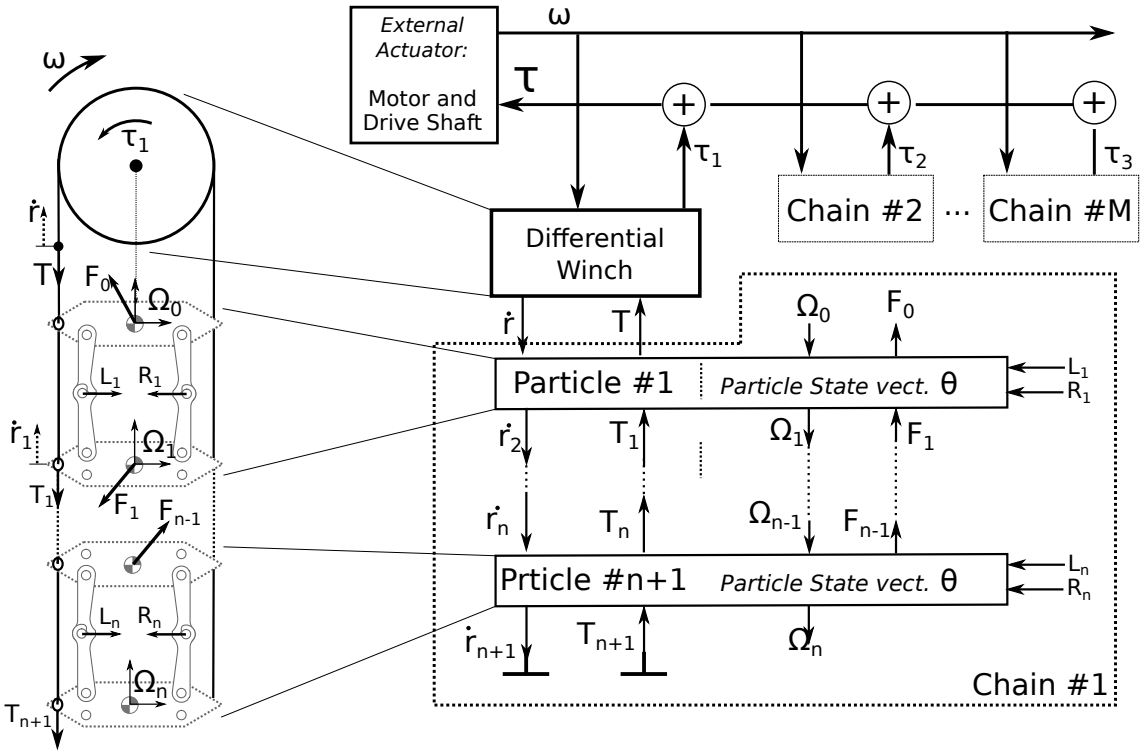


Figure 4.5: Components of an SSD and their mechanical coupling.

A chain is a planar multibody system composed of rigid bodies (links) and compliant elements (tendons). We initially only focus on the motion of rigid bodies, regardless of the presence of compliant elements. Subsequently, we model the effects that tendons produce on the system by considering the forces caused by their interaction with the eyelets.

The dynamics of the rigid bodies must eventually satisfy the Newton-Euler (NE) equation $\boldsymbol{M}_i\ddot{\mathbf{c}}_i = \mathbf{F}_i$, where $\boldsymbol{M}_i$, $\ddot{\mathbf{c}}_i$ and $\mathbf{F}_i$ are *mass*, *acceleration*, and *forces* applied to the center of mass of the i-th body. As the links of a chain are interconnected through revolute joints, we adopt *generalized coordinates* to indicate the relative angular displacement θ_i between pairs of adjoining bodies $(i-1)$ and i. In this way, the position $\mathbf{c}_i$, the velocity $\dot{\mathbf{c}}_i$ and the acceleration $\ddot{\mathbf{c}}_i$ of the center of mass of the body i can be defined as functions of the generalized coordinates $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_i]$ $(\mathbf{c}_i = \mathbf{c}_i(\boldsymbol{\theta}))$ and plugged into the NE equation.

As links are interconnected via revolute joints, their motion $\mathbf{c}_i(\boldsymbol{\theta})$ is subject to kinematic constraints. For example, the kinematic loop that the six links of a particle form, constrains the motion of link i which is affected by the motion of the links interconnected to i. When analyzing the dynamics of the system, the kinematic constraints also limit the effect of the force $\mathbf{F}_i$ applied to the link i, which can only produce motions that comply with the kinematic constraints. To efficiently model these situations, we first derive a kinematic model of the system to define the admissible spatial arrangements of the i-th body, namely the function $\mathbf{c}_i = \mathbf{c}_i(\boldsymbol{\theta})$ and its derivatives; then we impose the NE equation to analyze the dynamics of the system.

Kinematic Model. The rigid bodies (links) are subject to three types of kinematic

constraints which limit their relative motion and constrain the position of their centers of mass $\mathbf{c}_i$: *loop constraints*, *joint limit constraints*, and *rotational friction constraints*:

1. **Loop constraints** limit the relative motions of links of a chain due to the fact that links are mechanically joint into a loop. For example, considering the six links outlining a particle (Fig. 4.6), the relative angular displacement θ_i of links $(i-1)$ and i affects the status of the other five links. This implies a set of holonomic constraints that can be expressed in the form $\Phi^c(\theta_1, \cdots, \theta_n) = 0$, but for which it is not possible to derive equations of motion in a closed form [102].

2. **Joint limit constraints** limit the relative rotation of links connected by a revolute joint. The angle θ_i, enclosed by the links, must lie within a predefined angular range $\theta_{i,min} \leq \theta_i \leq \theta_{i,max}$. This non-holonomic constraint can be expressed as $\Phi^u(\theta_1, \theta_2, \cdots, \theta_n, \dot{\theta}_1, \cdots, \dot{\theta}_n) \leq 0$ which is also in an implicit form.

3. **Rotational friction constraints** limit the angular velocities of revolute joints because of the dynamic friction at the joints. This affects the relative angular velocity of adjoining links and can be expressed in the form $\Phi^f(\theta_1, \cdots, \theta_n) \leq 0$. However, as it is easier to formulate this type of non-holonomic constrains in terms of torques τ counteracting the relative rotation of adjoining links ($\tau = -k\dot{\theta}_i$), we consider these constraints as part of the dynamic model.

Loop (1.) and joint limit constraints (2.) (or, more in general, kinematic constrains) are expressed in an implicit form (e.g., $\Phi(\theta_1, \cdots, \theta_n) = 0$), from which it not possible to derive kinematic equations of the centers of mass $\mathbf{c}_i$ in an explicit form [102] (e.g., $\mathbf{c}_i = \mathbf{c}_i(\boldsymbol{\theta}, \Phi)$). To overcome this situation, we first derive the kinematics of an equivalent *unconstrained system* [18] for which $\mathbf{c} = \mathbf{c}(\theta)$ can be explicitly defined. Subsequently, as the motion of each body $\mathbf{c}_i = \mathbf{c}_i(\boldsymbol{\theta})$ must satisfy the kinematic constraints (e.g., $\Phi(\theta_1, \cdots, \theta_n) = 0$), a set of *reaction forces* is derived and applied to body i, such that these forces produce the same effects caused by the kinematic constraints. For example, reaction forces are derived to impede that the rotation of links violates the *joint limit constraints* (2.).

Dynamic Model. The dynamic model extends the kinematic model by defining the relationships among forces, masses and inertia which cause motion. Specifically, the following forces are taken into account:

1. **Gravitational** and **inertial** forces, due to gravity and acceleration.

2. **Actuation** forces, caused by tendons and built-in actuators acting on particles in the chain.

3. **Reaction** forces due to the three kinematic constraints listed above.

Summary of the Modeling Methodlogy. Our modeling methodology can be summarized as follows:

I. We first focus on the kinematic model. This includes:

   (a) Definition of an *unconstrained system* for which kinematic equations can be derived in an explicit form;

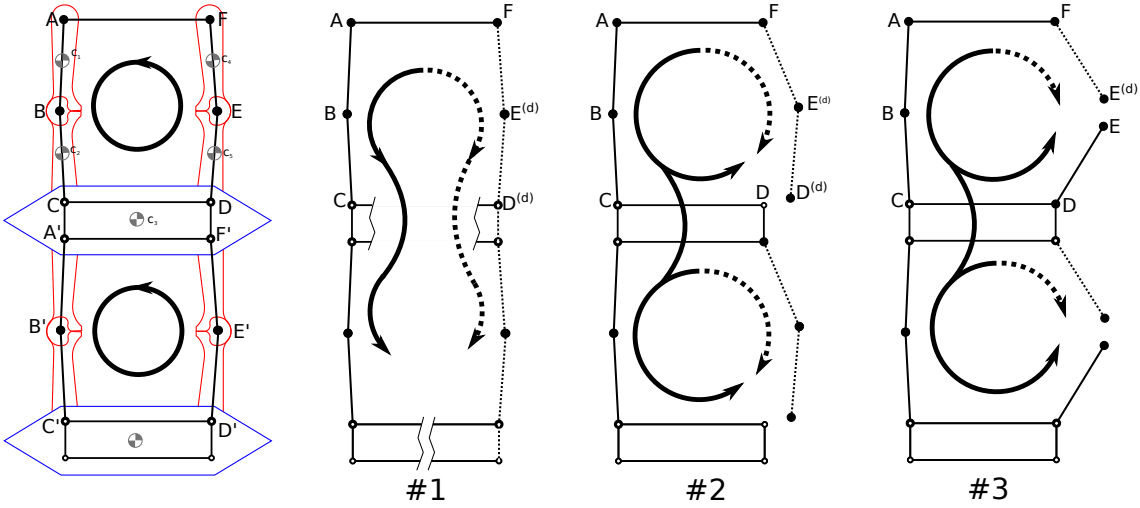


Figure 4.6: Possible unconstrained configurations.

(b) Definition of the kinematic constraints (in implicit form);

II. We extend the kinematics by applying the NE equation to obtain the dynamic model. This requires:

(a) Derivation of reaction forces that produce the same effect of the kinematic constraints;

(b) Derivation of the actuation forces;

(c) Derivation of gravitational and inertial forces.

Before proceeding with the model formulation, we introduce a set of assumptions to support our modeling approach.

4.3.4 Assumptions

A set of assumptions is defined here to facilitate the subsequent model formulation:

1. Ultimately, the dynamic model is applied to predict the behavior of the system over the time period in between consecutive configurations $Yc \rightarrow Y_{c+1}$. As this period is presumably short, constraint stabilization [7] is not necessary. .
2. The head of the chain is fixed to an inertial reference frame.
3. The backlash at the hinges is negligible.
4. The static Amonton-Coulomb friction model is sufficiently accurate to model the friction between tendon and eyelet, and also the friction at the joints.
5. Eyelets are considered dimensionless points through which tendons pass.
6. The internal friction of differential winches is negligible, therefore the two tensions applied to the head particle of the chain are equivalent.
7. Tendons are massless and inextensible.

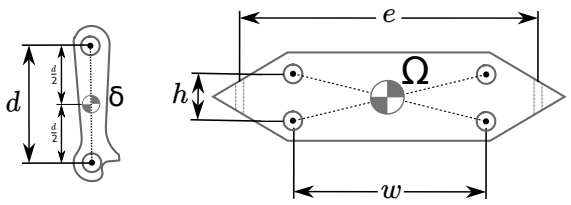


Figure 4.7: Types of links composing a chain: 2-joint link δ and 4-joint link Ω. Centers of mass and dimensions are indicated.

4.3.5 Kinematic Model Formulation

The kinematic model takes into account the kinematics of the rigid bodies and the constraints which limit the DOFs of the system. We first analyze the kinematics of an equivalent unconstrained system, where the loop constraints are temporarily disregarded. This results in a tree-type topology, where the kinematic equations describing the motion of each independent branch are derived with respect to joint coordinates (i.e., angular displacement of adjoining links). To subsequently model the mutual dependencies among branches, *loop constraint* equations define the relationships among joint coordinates of different branches. In addition, as revolute joints constrain the relative rotation of the interconnected links, a set of inequalities is introduced to appropriately model their effect.

A chain consists of rigid links interconnected through revolute joints, which form a regular and symmetric topology. We distinguish between two types of links, as Fig. 4.7 shows: δ-type links are 2-joint links whose center of mass coincides with the mid-point of the segment between the joints; Ω-type links are 4-joint links that have a regular geometry and thus their center of mass is the centroid of the four joints. As position and velocity of the center of mass of a link can be uniquely defined with respect to its joints, we refer in the following to the kinematics of the joints without loss of generality.

Unconstrained system

The links composing a chain form a planar regular topology, as shown in Fig. 4.6. An unconstrained system is essentially a spanning-tree [80,103] where loops (e.g., ABCDEFA) are temporarily opened. The unconstrained branches of a spanning-tree allow us to formulate the kinematic equations in an explicit form $\boldsymbol{c} = \boldsymbol{c}(\theta)$, where the vector $\boldsymbol{c} = [c_1, \cdots, c_i]$ indicates the center of mass of each link i and the vector $\theta = [\theta_1, \cdots, \theta_i]$ represents the angular displacements of adjoining links.

For the system at hand, we identify three basic alternatives for "opening" the loop closure, as shown in Fig. 4.6. In the first case #1, the unconstrained system is obtained by vertically cutting the Ω-type links. This results in two independent branches (e.g., ABC and $\mathrm{ED}^{(D)}\mathrm{F}^{(D)}$).
In the second case #2, the unconstrained system is obtained by disconnecting a δ-type link from the Ω-type link, specifically the joint D is disconnected. When applied to all the particles in the chain, this results in a tree-type topology where, for instance, the branches ABCD and $\mathrm{AFE}^{(D)}\mathrm{D}^{(D)}$ are independent of each other. Because of the symmetry of the chain, the disconnection of any other joint such as A or C or F is geometrically equivalent to the second case #2 depicted in Fig. 4.6.

In the third case #3, the joint E between the δ-type links is disconnected. An equivalent result can be obtained by disconnecting the joint B. As the second #2 and third #3 solutions lead to similar tree-type topologies, in the following we focus on the second case #2, as it results in a set of simpler kinematic equations.

The removed kinematic constraints need to be later reintroduced through additional equations, which model the mutual interaction among branches, namely define how the motion of a branch affects other branches. We distinguish between *independent* and *driven* branches. In Fig. 4.6, we indicate with $^{(D)}$ (e.g., $\mathrm{D}^{(D)}$) the joints that belong to a driven branch. Constraint equations condition the motion of driven branches, which are subject to the motion of independent branches.

Loop Constraints: Modeling approaches

In Papers $\boxed{C}$ and $\boxed{D}$ we propose two modeling approaches called *chain-oriented* and *particle-oriented*, which refer to the cases #1 and #2 of Fig. 4.6 and which are summarized below. In both cases, the kinematic constraint equations are derived referring to the motion of the joints (i.e., the vertices of the topology draft in Fig. 4.6) instead of considering the motion of the centers of mass, which would lead to less intuitive equations. Also, these equations are formulated with respect to relative joint coordinates (i.e., angular displacement of adjoining links). This does not limit the generality of the solution because position and motion of the centers of mass, which are necessary to formulate the dynamic model, can be directly inferred from position and motion of the joints of the links. Fig. 4.7 shows link dimensions and relative positions of centers of mass and joints.

Chain-oriented approach (Paper $\boxed{C}$).
The chain is modeled as a whole and the unconstrained system results from "splitting" the chain along the vertical line of symmetry into two branches $P = [P_1\ P_2\ \cdots\ P_{3n}]$ and $Q = [Q_1\ Q_2\ \cdots\ Q_{3n}]$, as shown in Fig. 4.8a. Only the top link of the chain is not split as it defines the relative position of the two branches and does not preclude the derivation of a closed-form equation. As the top link is the sole non-moving part, we refer to it as an inertial frame of reference.

The elements $\mathbf{p}_i$ and $\mathbf{q}_i$ in Fig. 4.8a are versors (unit vectors) associated to the links of each branch, sequentially enumerated starting from the top. Joint coordinates θ_i indicate the relative angular displacement of consecutive links (e.g., $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$), as indicated in Fig. 4.8a. This means that $\mathbf{p}_i = \left[\cos\sum_{j=1}^{i}\theta_j\ \ \sin\sum_{j=1}^{i}\theta_j\right]$. As in the unconstrained representation the two branches P and Q are free to move independently, two distinct vectors of relative joint coordinates $\boldsymbol{\theta}_P$ and $\boldsymbol{\theta}_Q$ are defined for P and Q.

With a slightly different notation than the one used in paper $\boxed{C}$, the following kinematic equation maps the absolute position of the joint P_i and the relative joint coordinates $\boldsymbol{\theta}_P = [\theta_1 \cdots \theta_i]$, of the unconstrained branch P:

$$P_i(\boldsymbol{\theta}_P) \equiv \begin{bmatrix}\mathbf{p}_1 & \cdots & \mathbf{p}_i\end{bmatrix} \cdot \begin{bmatrix} d_1 \\ \vdots \\ d_i \end{bmatrix} = \begin{bmatrix}\mathbf{p}_1 & \cdots & \mathbf{p}_i\end{bmatrix} \cdot \mathbf{d}_i \tag{4.7}$$

where the element of the vector $\mathbf{d}_i$ is the length of the corresponding link. Considering the offset between the point P_1 and Q_1, namely the distance $w = \overline{P_1 Q_1}$, a similar equation

holds for the branch Q:

$$Q_i(\boldsymbol{\theta}_Q) \equiv \begin{bmatrix}\mathbf{q}_1 & \cdots & \mathbf{q}_i\end{bmatrix} \cdot \mathbf{d}_i + \begin{bmatrix} w \\ 0 \end{bmatrix} \tag{4.8}$$

Let the versor $\hat{\mathbf{p}}_i = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \cdot \mathbf{p}_i$ be the 90° counter-clockwise rotation of the versor $\mathbf{p}_i$. The two versors $\mathbf{p}_i$ and $\hat{\mathbf{p}}_i$ define a frame of reference associated to each link with origin at the center of mass of the link and orientation $\sum_{j=1}^{i} \theta_j$. For instance, in Fig. 4.8a the frame of reference Ω_i is defined for the Ω-type link i. The positions of joints and eyelets of link i are defined with respect to the local frame Ω_i. In this way, the forces later introduced with the dynamic model which act on link i, can be defined with respect to the local frame Ω_i, without loss of generality.

Loop constraints bind the relative motion of the branches P and Q. A set of constraint equations $\Phi^c(\boldsymbol{\theta}_P, \boldsymbol{\theta}_Q) \equiv \mathbf{0}$ defines the mutual influence of these two branches. In particular, the time derivative $\dot{\Phi}^c \equiv \mathbf{0}$ (which is also null) indicates the co-variation of the vectors $\boldsymbol{\theta}_P$ and $\boldsymbol{\theta}_Q$. As the block diagram in Fig. 4.8a suggests, given the vector $\boldsymbol{\theta}_P$ and the constraint Φ^c, the position of the generic frame Ω_i is uniquely defined.
The constraint equation $\Phi^c \equiv \mathbf{0}$ can be obtained considering for each cut link i that:

- the cut parts must be aligned, thus the versors $\mathbf{p}_i$ and $\mathbf{q}_i$ must be parallel;
- the relative distance between the cut parts is constant, that is $|\overline{P_i Q_i}| = w$.

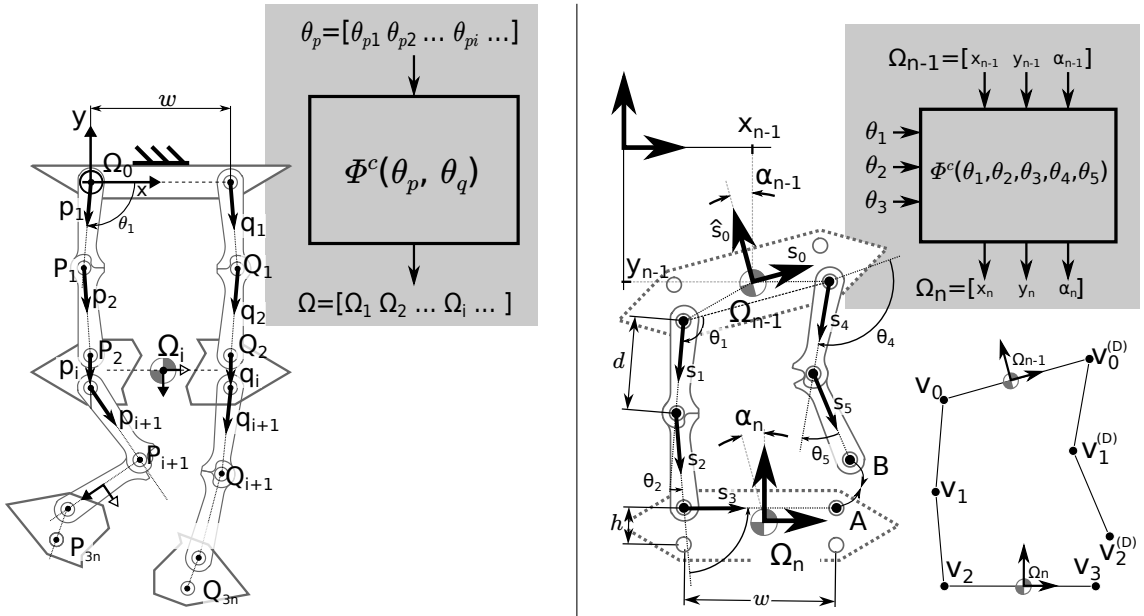
This results in the following equations that must be satisfied for $\forall i = 3\ t \in \mathbb{N}, 0 < t \leq \frac{N}{3}$ (with N the number of links in a branch):

$$\Phi_i^c(\boldsymbol{\theta}_P, \boldsymbol{\theta}_Q) = \begin{cases} \sum_{j=1}^{i} \theta_{P_i} - \sum_{j=1}^{i} \theta_{Q_i} & = 0 \\ P_i + \hat{\mathbf{p}}_i \cdot w - Q_i & = 0 \end{cases} \tag{4.9}$$

The above equations are in an implicit form, which does not allow us to invert the function Φ_i^c and hence to derive $\boldsymbol{\theta}_Q = (\Phi_i^c)^{-1}(\boldsymbol{\theta}_P)$. To overcome this situation, the above equations are introduced in the dynamic model to infer an equivalent set of forces which applied to the cut links satisfy the loop constraints.

Particle-oriented approach (Paper D).
Referring to Fig. 4.8b, a single particle is modeled as an independent element. As the chain results from the concatenation of consecutive particles having their top and bottom bases coincident, the kinematics of the n-th particle depends recursively on the kinematics of the $(n-1)$-th particle [103]. The top base of particle n has an associated frame of reference Ω_{n-1}, which coincides with the bottom base of particle $n-1$. Similarly, the bottom base of particle n has an associated frame Ω_n which is a function of Ω_{n-1} and the internal state of the particle, namely the joint coordinates θ_1, θ_2 and θ_3 (the joint coordinates θ_4 and θ_5 are *driven* coordinates which depend on θ_1, θ_2 and θ_3). Position and orientation of $\Omega_n = [x_n\ y_n\ \alpha_n]$ are relative to Ω_{n-1}, where $(x_n\ y_n)$ is the linear displacement and α_n is the angular displacement (the angle enclosed by consecutive Ω-type links of a particle is the defined in Sec. 4.2.4).

The unconstrained system is obtained by disconnecting the six links forming a particle at one selected joint, as shown in Fig. 4.8b. If we consider Ω_{n-1} as the local reference frame, this results in two branches $\mathbf{s}_0, \mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3$ and $\mathbf{s}_0, \mathbf{s}_4, \mathbf{s}_5$. A link-associated versor $\mathbf{s}_i$



(a) Chain-oriented modeling approach. **(b)** Particle-oriented modeling approach.

Figure 4.8: The two alternative formulations presented in Papers C and D.

indicates the orientation of each link i and defines a frame of reference integral to each link having axes $\mathbf{s}_i$ and $\hat{\mathbf{s}}_\mathbf{i} = \left[\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right] \cdot \mathbf{s}_i$ (i.e., 90° counter-clockwise rotation of $\mathbf{s}_i$). Special cases are the frame of reference Ω_{n-1} associated to the top base of the particle, having origin at its center of mass and coinciding with $(\mathbf{s}_0, \hat{\mathbf{s}}_\mathbf{0})$, and the frame of reference Ω_n associated to the bottom base of the particle, also having origin at the center of mass and coinciding with $(\mathbf{s}_3, \hat{\mathbf{s}}_\mathbf{3})$. The frames Ω_{n-1} and Ω_n have *absolute* coordinates $\boldsymbol{c}_{n-1}$ and $\boldsymbol{c}_n$ (positions of the centers of mass) and absolute angular displacements $\alpha_{c_{n-1}}$ and α_{c_n}.

Joint coordinates θ_i indicate the relative angular displacement of consecutive links (e.g., $\mathbf{s}_i$ and $\mathbf{s}_{i+1}$), with the exception that θ_4 is the angle enclosed by $\mathbf{s}_0$ and $\mathbf{s}_4$, as shown in Fig. 4.8b. As the block in Fig. 4.8b suggests, when analyzing the unconstrained system, the frame Ω_n depends on the frame Ω_{n-1} and on the angles θ_1, θ_2 and θ_3.

A constraint function $\Phi^c(\boldsymbol{\theta}) \equiv 0$ can be derived to define the relationship among the θ_i elements of the vector $\boldsymbol{\theta}$. To this end, we first provide the following equation, which indicates the positions of the vertices $\mathbf{v}_j$ (see Fig. 4.8b) as a function of the joint coordinates $\mathbf{v} = f(\boldsymbol{\theta})$, and then impose the constraint $\mathbf{v}_3 \equiv \mathbf{v}_2^{(D)}$:

$$\begin{bmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \\ \mathbf{v}_0^{(D)} \\ \mathbf{v}_1^{(D)} \\ \mathbf{v}_2^{(D)} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & 0 & 0 & 0 & 0 \\ \mathbf{I} & \mathbf{I} & 0 & 0 & 0 \\ 0 & \mathbf{I} & \mathbf{I} & 0 & 0 \\ 0 & 0 & \mathbf{I} & \mathbf{I} & 0 \\ \mathbf{I} & 0 & 0 & 0 & 0 \\ \mathbf{I} & 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & \mathbf{I} & \mathbf{I} \end{bmatrix} \cdot \left(\begin{bmatrix} -\frac{w}{2} & & & & \\ 0 & d & & & \\ 0 & 0 & d & & \\ 0 & 0 & 0 & w & \\ w & & & & \\ 0 & 0 & 0 & 0 & d \\ 0 & 0 & 0 & 0 & 0 & d \end{bmatrix} \begin{bmatrix} \mathbf{s}_0 \\ \mathbf{s}_1 \\ \mathbf{s}_2 \\ \mathbf{s}_3 \\ \mathbf{s}_4 \\ \mathbf{s}_5 \end{bmatrix} + \begin{bmatrix} -\frac{h}{2} \cdot \hat{\mathbf{s}}_\mathbf{0} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right) \tag{4.10}$$

In the above equation the joint coordinates θ_i are not explicitly indicated, but they are inherent to each versor $\mathbf{s}_i = \mathbf{s}_i(\boldsymbol{\theta})$ and their 90°-counterclockwise pair $\hat{\mathbf{s}}_\mathbf{i} = \hat{\mathbf{s}}_\mathbf{i}(\boldsymbol{\theta})$. The dimensions w and h are width and height of a Ω-type link, while d is the length of a δ-type link, as indicated in Fig. 4.7. The above equation highlights the mutual dependencies of vertices, which are recursively defined – e.g., $\mathbf{v}_1$ depends on $\mathbf{v}_0$.

The loop-constraints $\Phi^c(\boldsymbol{\theta})$ are satisfied when the two joints $\mathbf{v}_3$ and $\mathbf{v}_2^{(D)}$ coincide, namely $\Phi^c(\boldsymbol{\theta}) = \mathbf{v}_3 - \mathbf{v}_2^{(D)} \equiv 0$. This relationship can be inferred from Eq. 4.10 as:

$$\Phi^c(\boldsymbol{\theta}) = \underbrace{-\frac{w}{2}\mathbf{s}_0 - \frac{h}{2}\hat{\mathbf{s}}_\mathbf{0} + d\mathbf{s}_1 + d\mathbf{s}_2 + w\mathbf{s}_3}_{\mathbf{v}_3} - \underbrace{\left(\frac{w}{2}\mathbf{s}_0 - \frac{h}{2}\hat{\mathbf{s}}_\mathbf{0} + d\mathbf{s}_4 + d\mathbf{s}_5\right)}_{\mathbf{v}_2^{(D)}} = 0 \tag{4.11}$$

The above equation implies that a variation of any angle $\theta_{1,2,3}$ causes a corresponding variation of the angles θ_4 and θ_5 of the driven branch, such that $\mathbf{v}_3 \equiv \mathbf{v}_2^{(D)}$. The time derivative of the above equation, which also needs to be null, characterizes how a variation of any θ_i affects the relative positions of $\mathbf{v}_3$ and $\mathbf{v}_2^{(D)}$:

$$\frac{d}{dt}\Phi^c(\boldsymbol{\theta}) = \underbrace{\frac{\partial \Phi^c(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}}_{\boldsymbol{J}^c} \cdot \boldsymbol{\theta} = \frac{\partial \mathbf{v}_3}{\partial \boldsymbol{\theta}} \cdot \dot{\boldsymbol{\theta}} + \frac{\partial \mathbf{v}_2^{(D)}}{\partial \boldsymbol{\theta}} \cdot \dot{\boldsymbol{\theta}} = 0 \tag{4.12}$$

From the Jacobian matrix $\boldsymbol{J}^c = \frac{\phi_i^u(\boldsymbol{\theta})}{d\boldsymbol{\theta}}$, it is possible to infer a set of *reaction forces* that counteracts the external forces and satisfies the loop constraint. More precisely, in the dyanmic model the *reaction forces* are derived in form of Lagrange's multipliers [66] applied to $\boldsymbol{J}^c$. Equation 4.12 yields the following Jacobian $\boldsymbol{J}^c$, which can be considered constant for a short time interval (assumption 1):

$$\boldsymbol{J}^c = \begin{bmatrix}(d \cdot (\hat{\mathbf{s}}_\mathbf{1} + \hat{\mathbf{s}}_\mathbf{2}) + w \cdot \hat{\mathbf{s}}_\mathbf{3}) & (d \cdot \hat{\mathbf{s}}_\mathbf{2} + w \cdot \hat{\mathbf{s}}_\mathbf{3}) & (w \cdot \hat{\mathbf{s}}_\mathbf{3}) & (-d \cdot (\hat{\mathbf{s}}_\mathbf{4} + \hat{\mathbf{s}}_\mathbf{5})) & (-d \cdot \hat{\mathbf{s}}_\mathbf{5})\end{bmatrix} \tag{4.13}$$

where, with respect to Eq. 4.11, $\mathbf{s}_0$ disappears because it is independent of any θ_i.

Center of Mass. The *absolute* position $\mathbf{c}_i$ and the angular displacement α_{c_i} of the center of mass of the rigid body i can be derived from Eq. 4.7 (chain-oriented approach) or from Eq. 4.10 (particle-oriented approach) in the form $\mathbf{c} = \mathbf{c}(\boldsymbol{\theta})$.

As in the remainder we mainly focus on the particle-oriented approach (for details on the chain-oriented approach we refer to paper $\boxed{C}$), we report here the complete formulation of equation $\mathbf{c} = \mathbf{c}(\boldsymbol{\theta})$:

$$\begin{bmatrix}\mathbf{c}_1 \\ \alpha_{c_1} \\ \hline \mathbf{c}_2 \\ \alpha_{c_2} \\ \hline \mathbf{c}_3 \\ \alpha_{c_3} \\ \hline \mathbf{c}_4 \\ \alpha_{c_4} \\ \hline \mathbf{c}_5 \\ \alpha_{c_5}\end{bmatrix} = \begin{bmatrix}\frac{d}{2}\mathbf{s}_1 \\ \theta_1 \\ \hline d\mathbf{s}_1 + \frac{d}{2}\mathbf{s}_2 \\ \theta_1 + \theta_2 \\ \hline d\,(\hat{\mathbf{s}}_\mathbf{1} + \hat{\mathbf{s}}_\mathbf{2}) + \frac{w}{2}\mathbf{s}_3 - \frac{h}{2}\hat{\mathbf{s}}_\mathbf{3} \\ \theta_1 + \theta_2 + \theta_3 \\ \hline \frac{d}{2}\mathbf{s}_4 \\ \theta_4 \\ \hline d\mathbf{s}_4 + \frac{d}{2}\mathbf{s}_5 \\ \theta_4 + \theta_5\end{bmatrix} + \begin{bmatrix}-\frac{w}{2}\mathbf{s}_0 - \frac{h}{2}\hat{\mathbf{s}}_\mathbf{0} \\ \alpha_{n-1} \\ -\frac{w}{2}\mathbf{s}_0 - \frac{h}{2}\hat{\mathbf{s}}_\mathbf{0} \\ \alpha_{n-1} \\ -\frac{w}{2}\mathbf{s}_0 - \frac{h}{2}\hat{\mathbf{s}}_\mathbf{0} \\ \alpha_{n-1} \\ \frac{w}{2}\mathbf{s}_0 - \frac{h}{2}\hat{\mathbf{s}}_\mathbf{0} \\ \alpha_{n-1} \\ \frac{w}{2}\mathbf{s}_0 - \frac{h}{2}\hat{\mathbf{s}}_\mathbf{0} \\ \alpha_{n-1}\end{bmatrix} \tag{4.14}$$

The above equation is recursively derived with respect to the reference frame Ω_{n-1} and refers to particle n, although for brevity of notation any index n is omitted (i.e., $\boldsymbol{c}_1$ should be read as $\boldsymbol{c}_{1_n}$). Similarly, the parameter $\boldsymbol{\theta}$ is omitted for the versors $\mathbf{s}_j = \mathbf{s}_j(\boldsymbol{\theta})$ and $\hat{\mathbf{s}}_\mathbf{j} = \hat{\mathbf{s}}_\mathbf{j}(\boldsymbol{\theta})$.
As for each particle n, the bottom link Ω_n has origin in $\mathbf{c}_3$ and has absolute angular displacement α_{c_3}, also the frame Ω_n is recursively defined with respect to Ω_{n-1}. This facilitates the formulation of the dynamic model as the position of eyelets and joints subject to external and internal forces can be defined with respect to the local frame Ω_n without loss of generality; likewise the external and the internal forces can be referred to the local frame Ω_n.

We complete the kinematic model deriving the velocities $\dot{\mathbf{c}}$ of the centers of mass from the above equation. This is later required to support the formulation of the dynamic model. The derivative of $\mathbf{c} = \mathbf{c}(\boldsymbol{\theta})$ can be expressed in the form $\dot{\mathbf{c}} = \boldsymbol{B}\dot{\boldsymbol{\theta}}$, where $\boldsymbol{B}$ is the Jacobian of the function $\mathbf{c}(\boldsymbol{\theta})$:

$$\begin{bmatrix} \dot{\mathbf{c}}_1 \\ \dot{\alpha}_{c_1} \\ \dot{\mathbf{c}}_2 \\ \dot{\alpha}_{c_2} \\ \dot{\mathbf{c}}_3 \\ \dot{\alpha}_{c_3} \\ \dot{\mathbf{c}}_4 \\ \dot{\alpha}_{c_4} \\ \dot{\mathbf{c}}_5 \\ \dot{\alpha}_{c_5} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{d}{2}\hat{\mathbf{s}}_\mathbf{1} & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ \frac{d}{2}\hat{\mathbf{s}}_\mathbf{2} + \hat{\mathbf{s}}_\mathbf{1} d & \frac{d}{2}\hat{\mathbf{s}}_\mathbf{2} & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ \frac{h}{2}\mathbf{s}_3 + \frac{w}{2}\hat{\mathbf{s}}_\mathbf{3} + d\,(\hat{\mathbf{s}}_\mathbf{1} + \hat{\mathbf{s}}_\mathbf{2}) & \frac{h}{2}\mathbf{s}_3 + \frac{w}{2}\hat{\mathbf{s}}_\mathbf{3} + \hat{\mathbf{s}}_\mathbf{2} d & \frac{h}{2}\mathbf{s}_3 + \frac{w}{2}\hat{\mathbf{s}}_\mathbf{3} & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{d}{2}\hat{\mathbf{s}}_\mathbf{4} & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{d}{2}\hat{\mathbf{s}}_\mathbf{5} + \hat{\mathbf{s}}_\mathbf{4} d & \frac{d}{2}\hat{\mathbf{s}}_\mathbf{5} \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}}_{\boldsymbol{B}} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \end{bmatrix} \tag{4.15}$$

For a short time interval (assumption 1), the matrix $\boldsymbol{B}$ of Eq. 4.14 is considered constant with respect to $\boldsymbol{\theta}$. This establishes a linear relationship between a variation of the joint coordinates $\boldsymbol{\theta}$ and a variation of the positions of the center of mass $\mathbf{c}$.

Similarly, we can derive the accelerations $\ddot{\mathbf{c}} = \dot{\boldsymbol{B}}\dot{\mathbf{c}} + \boldsymbol{B}\ddot{\mathbf{c}}$. However, considering that the term $\dot{\boldsymbol{B}}\dot{\mathbf{c}}$ leads to numerical instability [100], the dynamic model is later formulated considering an approximation of $\ddot{\mathbf{c}}$, thus we omit the formal derivation of $\ddot{\mathbf{c}}$.

Joint Limit Constraints

Revolute joints are designed to limit the relative rotation of adjoining links. Each element θ_i of the vector $\boldsymbol{\theta}$ is bound within a predefined angular range $\theta_i \in [\theta_{\min_i}, \theta_{\max_i}]$. Fig. 4.9 depicts for each joint i the angular range of the corresponding angle θ_i. As in the unconstrained system (particle-oriented approach) one joint is disconnected, the vector $\boldsymbol{\theta}$ contains only five elements corresponding to $[\theta_1 \cdots \theta_5]$. In order to characterize the angular limit at each joint, the sixth angle θ_6 is derived from the other five angles according to the following equation:

$$\theta_6 = -(\theta_1 + \theta_2 + \theta_3) + \theta_4 + \theta_5 + \frac{\pi}{2} \tag{4.16}$$

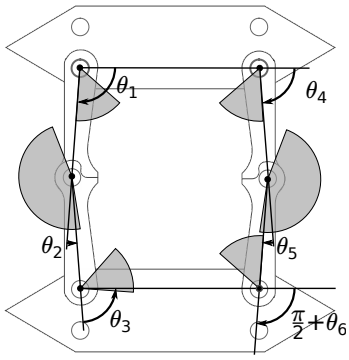


Figure 4.9: Revolute joint angular limits.

The following inequalities define for each joint the angular limit that the mechanical design imposes:

$$\boldsymbol{\theta}_{\min} = \begin{bmatrix} -\frac{\pi}{2} - \delta_1 \\ -\delta_3 \\ \frac{\pi}{2} - \delta_1 \\ -\frac{\pi}{2} - \delta_2 \\ -\delta_4 \\ -\delta_1 \end{bmatrix} \leq \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \end{bmatrix} \leq \begin{bmatrix} -\frac{\pi}{2} + \delta_2 \\ \delta_4 \\ \frac{\pi}{2} + \delta_2 \\ -\frac{\pi}{2} + \delta_1 \\ \delta_3 \\ +\delta_2 \end{bmatrix} = \boldsymbol{\theta}_{\max} \tag{4.17}$$

referring to the prototype at hand: $\delta_1 \approx 0.0769$, $\delta_2 \approx 0.8599$, $\delta_3 \approx 2.7669$, $\delta_4 \approx 0.1591$.

The above inequalities define a set of *unilateral* constraints that are satisfied as long as each angle θ_i lies strictly within the defined range, namely $\theta_{\min_i} < \theta_i < \theta_{\max_i} \forall i \in [1..6]$. When any θ_i reaches one of the two limits ($\theta_i = \theta_{\max_i}$ or $\theta_i = \theta_{\min_i}$) an auxiliary constraint equation needs to be temporarily introduced in the kinematic model, to ensure that θ_i does not exceed the admissible limit. To clarify this concept, let us consider the case of θ_i reaching the upper limit $\theta_{\max_i}$ (analogue considerations apply for θ_i reaching $\theta_{\min_i}$). There are two cases to be considered when $\theta_i \equiv \theta_{\max_i}$:

1. The forces acting on the link i make the corresponding angular velocity $\dot{\theta}_i \leq 0$ non-positive, thus the link i tends to stand still or to move away from the maximum limit; in both cases the condition $\theta_i \leq \theta_{\max_i}$ is eventually satisfied.

2. The forces acting on the link i make the corresponding angular velocity $\dot{\theta}_i > 0$ positive, thus the link i tends to violate the maximum limit $\theta_i \leq \theta_{\max_i}$ as θ_i increases; in this case, an auxiliary constraint equation is required to ensure that $\theta_i = \theta_{\max_i}$, namely that $\dot{\theta}_i = 0$.

The following constraint equations come *temporarily* into play for the kinematic model as soon as the angular displacement θ_i reaches one of the two limits $\theta_{\max_i}$ or $\theta_{\min_i}$, and as long as the angular velocity $\dot{\theta}_i$ is positive or negative, respectively:

$$\begin{cases} \text{if } \theta_i \equiv \theta_{\min_i} \text{ and } \dot{\theta}_i < 0, & \Phi_i^u(\boldsymbol{\theta}) = \theta_i - \theta_{\min_i} \equiv 0 \\ \text{if } \theta_i \equiv \theta_{\max_i} \text{ and } \dot{\theta}_i > 0, & \Phi_i^u(\boldsymbol{\theta}) = \theta_i - \theta_{\max_i} \equiv 0 \end{cases} \tag{4.18}$$

By computing the time derivative of $\Phi_i^u(\boldsymbol{\theta})$, we notice that also the angular velocity $\dot{\theta}_i$ needs to be null in order to have the joint limit constraint satisfied:

$$\frac{\phi_i^u(\boldsymbol{\theta})}{dt} = \frac{\phi_i^u(\boldsymbol{\theta})}{d\boldsymbol{\theta}}\dot{\boldsymbol{\theta}} = \dot{\theta}_i \equiv 0 \tag{4.19}$$

which applies for $i \in [1..5]$. A special case is considered for the sixth joint ($i = 6$) for which the derivative $\phi_6^u(\boldsymbol{\theta})$ results in:

$$\frac{\phi_6^u(\boldsymbol{\theta})}{dt} = \frac{\phi_6^u(\boldsymbol{\theta})}{d\boldsymbol{\theta}}\dot{\boldsymbol{\theta}} = -\dot{\theta}_1 - \dot{\theta}_2 - \dot{\theta}_3 + \dot{\theta}_4 + \dot{\theta}_5 \equiv 0 \tag{4.20}$$

Equation 4.19 and Eq. 4.20 are adopted in the dynamic model to infer the *reaction force* τ_i that applied to link i counteracts the external forces in order to satisfy the joint limit constraints. The *reaction forces* are derived in form of Lagrange's multipliers [66].
To this end, the Jacobian $\boldsymbol{J}_i^u = \frac{\phi_i^u(\boldsymbol{\theta})}{d\boldsymbol{\theta}}$ is concatenated to the Jacobian $\boldsymbol{J}^c$ of Eq. 4.13, and reaction forces are sought to satisfy the loop and the joint limit constraints simultaneously.

The following equation defines the Jacobian $\boldsymbol{J}_i^u$ for each $i \in [1..6]$:

$$\boldsymbol{J}_i^u = \frac{\phi_i^u(\boldsymbol{\theta})}{d\boldsymbol{\theta}} = \begin{cases} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}, & \text{if } i = 1 \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \end{bmatrix}, & \text{if } i = 2 \\ \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix}, & \text{if } i = 3 \\ \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix}, & \text{if } i = 4 \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix}, & \text{if } i = 5 \\ \begin{bmatrix} -1 & -1 & -1 & 1 & 1 \end{bmatrix}, & \text{if } i = 6 \end{cases} \tag{4.21}$$

4.3.6 Dynamic Model Formulation

Following the NE formulation, all the bodies (links) composing a chain must satisfy the following equation(s):

$$\boldsymbol{M}\ddot{\mathbf{c}} = \mathbf{F}^e + \mathbf{F}^i \tag{4.22}$$

where:

- $\boldsymbol{M} = \text{diag}(\cdots\ M_i\ M_i\ I_i\ \cdots)$ is a time-invariant diagonal matrix, which indicates for each body (link) i the *mass* M_i and *moment of inertia* I_i at the center of mass of the body.
- $\ddot{\mathbf{c}} = [\cdots \ddot{c}_{x_i}\ \ddot{c}_{x_i}\ \ddot{\alpha}_{c_i} \cdots]$ is a vector indicating *linear* and *angular* accelerations with respect to the center of mass of each body composing the system. It is the time derivative of Eq. 4.15. Each element of $\ddot{\mathbf{c}}$ matches a corresponding element in $\boldsymbol{M}$.
- $\mathbf{F}^e = [\cdots\ f_{x_i}^e\ f_{y_i}^e\ \tau_i^e\ \cdots]^\top$ defines the *external* forces and torques acting on the i-th link with respect to its barycenter (for example, actuation and gravitational forces); elements of the vector match corresponding elements in $\ddot{\mathbf{c}}$.
- $\mathbf{F}^i = [\cdots\ f_{x_i}^i\ f_{y_i}^i\ \tau_i^i\ \cdots]^\top$ includes the *internal* forces and torques that other bodies exert on the i-th body with respect to its barycenter (for example, the constraint forces that prevent the six links of a particle from disconnecting); elements of the vector $\mathbf{F}^i = [\cdots\ f_{x_i}^i\ f_{y_i}^i\ \tau_i^i\ \cdots]^\top$ match corresponding elements of $\ddot{\mathbf{c}}$.

Goal of the dynamic model is to predict the behavior of the particle chain subject to actuation forces. In the following, we refer to the particle-oriented approach to infer the joint coordinates $\boldsymbol{\theta}$ (see Sec. 4.3.5) that satisfy both the kinematic constraints and the above NE equation.
Eq. 4.14 defines the kinematics of the bodies composing the system with respect to the joint coordinates $\boldsymbol{\theta}$, in the form $\mathbf{c} = \mathbf{c}(\boldsymbol{\theta})$. In order to combine Eq. 4.14 and Eq. 4.22, the second derivative of the former is required. The first derivative of Eq. 4.14 is provided in Eq. 4.15 as $\dot{\mathbf{c}} = \boldsymbol{B}\dot{\boldsymbol{\theta}}$. The second deriveative of Eq. 4.14 can be obtained from the latter in the form $\ddot{\mathbf{c}} = \boldsymbol{B}\ddot{\boldsymbol{\theta}} + \dot{\boldsymbol{B}}\dot{\boldsymbol{\theta}}$. However, as the last term $\dot{\boldsymbol{B}}\dot{\boldsymbol{\theta}}$ leads to numerical instability [100] when the model is applied in simulation, we approximate $\ddot{\mathbf{c}}$ as follows:

$$\ddot{\mathbf{c}} = \frac{\dot{\mathbf{c}}_k - \dot{\mathbf{c}}_{k-1}}{\Delta t} \tag{4.23}$$

where $\dot{\mathbf{c}}_k$ and $\dot{\mathbf{c}}_{k-1}$ are the velocities observed at two consecutive points in time over the period Δt: $\dot{\mathbf{c}}_k \equiv \dot{\mathbf{c}}(t)$ and $\dot{\mathbf{c}}_{k-1} \equiv \dot{\mathbf{c}}(t - \Delta t)$. The above equation suggests that the future status of the system $\dot{\mathbf{c}}_k$ can be inferred, knowing the current status $\dot{\mathbf{c}}_{k-1}$.

We approximate Eq. 4.22 according to Eq. 4.23 and combine the result with Eq. 4.15:

$$\boldsymbol{MB}(\dot{\boldsymbol{\theta}}_k - \dot{\boldsymbol{\theta}}_{k-1}) = (\mathbf{F}^e + \mathbf{F}^i) \cdot \Delta t \tag{4.24}$$

in which we assume $\boldsymbol{B}$ to be constant over the time period $(t - \Delta t, t)$.
From the above equation the future status of the system $\dot{\boldsymbol{\theta}}_k$ is derived with respect to the current status $\dot{\boldsymbol{\theta}}_{k-1}$:

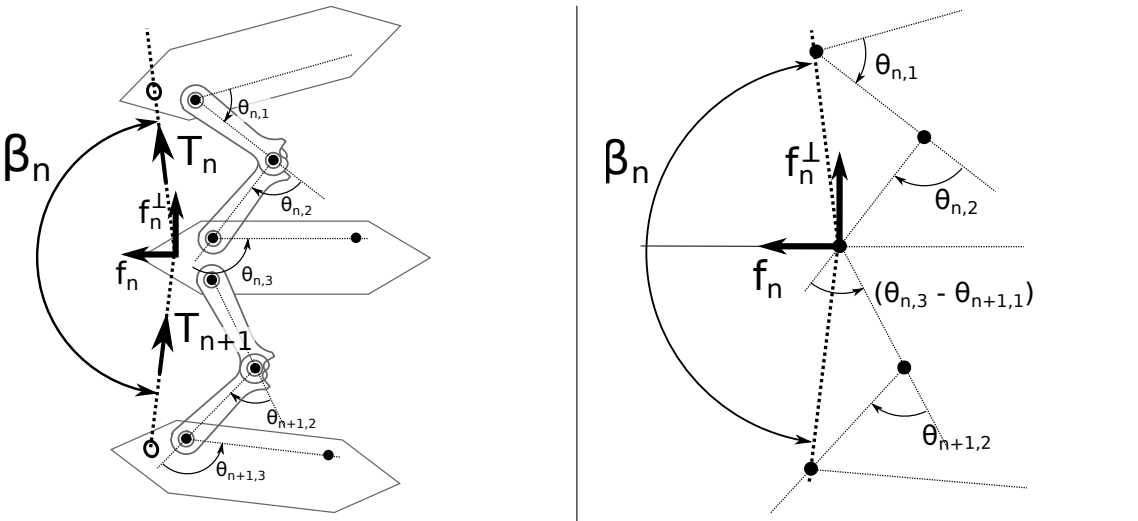
$$\dot{\boldsymbol{\theta}}_k = (\boldsymbol{B}^\top \boldsymbol{MB})^{-1} \boldsymbol{B}^\top (\mathbf{F}^e + \mathbf{F}^i) \cdot \Delta t + \dot{\boldsymbol{\theta}}_{k-1} \tag{4.25}$$

The above equation defines a fundamental relationship between the joint coordinates $\dot{\boldsymbol{\theta}}$ and the forces applied to the system. The latter result from the combination of external $\mathbf{F}^e$ and internal $\mathbf{F}^i$ forces. External forces can be further distinguished into *gravitational* and *actuation* forces; internal forces include the *reaction* forces due to the three types of constraint (loop, joint-limit and rotational-friction constraints) introduced in the kinematic model. External and internal forces are derived below.

External Forces $\mathbf{F}^e$

The force vector $\mathbf{F}^e$ in Eq. 4.24 represents the external linear forces and torques applied to the bodies composing a particle chain. In our system, $\mathbf{F}^e$ is characterized by gravitational and actuation forces. The latter derive from the interaction of tendons and eyelets and also from the action of the local actuators embedded in each particle responsible for unlocking one of the two folding sides. In the following, we analyze three cases.

I. **Gravitational Forces.** Equation 4.14 indicates the absolute angular displacement α_{c_i} of each link c_i composing a particle. To compute the effect of gravity on each link, we consider the relative orientation of the gravitational acceleration vector $\mathbf{g}$. The resulting forces are added to the force vector $\mathbf{F}^e$ of Eq. 4.25 which combines the effect of gravity on the system.



(a) Tension propagation model. **(b)** Gap between eyelet and joints neglected.

Figure 4.10: Tension propagation model.

II. **Tendon-Eyelet Interaction.** Tendons are assumed to be inextensible (assumption 7), thus, as long as they are in tension they act like rigid bodies exerting a tensile force between two points (i.e., consecutive eyelets). On the contrary, when the tendons get loose, namely their tension becomes negative, their compliance needs to be modeled. However, as a fundamental requirement to ensure the controllability of a tendon-driven system is that the tension applied to tendons must always be positive, our conclusion in case of negative tension is that the chosen actuation strategy is unstable and thus a different actuation strategy needs to be found. This dispenses us from explicitly modeling the compliance of loose tendons and at the same time provides valuable information necessary for optimal planning and control algorithms.

The compliance of the tendon is nevertheless implicitly modeled as the tendons adapt their path to the curvature of the chain. This affects the tension that propagates through the chain and that we model following an approach similar to the one proposed in Sec. 4.2.4. When the path of a tendon traversing a pinhole eyelet (assumption 5) deviates, the contact between tendon and eyelet originates two orthogonal forces $\mathbf{f}_n$ and $\mathbf{f}_n^\top$ both applied to the eyelet, as shown in Fig. 4.10a. The first force $\mathbf{f}_n$ is parallel to the line bisecting the two tendons departing from the eyelet; the second force $\mathbf{f}_n^\top$ is perpendicular to the first one and can be derived from the Coulomb's friction model $\mathbf{f}_n^\top = \mu \cdot \mathbf{f}_n$, where μ is the coefficient of static friction between tendon and eyelet as defined in Sec. 4.2.4.
Eq. 4.2.4 can be adapted to model the situation depicted in Fig. 4.10a, as follows:

$$T_{n+1} = |\mathbf{T_n} - \mathbf{f_n^\top}| = T_n\big(1 - \mu \cdot |cos(\beta_n)|\big) \tag{4.26}$$

The angle β_n shown in Fig. 4.10a depends on the relative positions of the three eyelets which the tendon traverses. Assuming the gap between the eyelet and the two closest hinges to be negligible (as in Fig. 4.10b), the following equation approximates the

angle β_n:

$$\beta_n = \pi + \theta_{n,3} - \theta_{n+1,1} - \frac{\theta_{n,1}}{2} - \frac{\theta_{n+1,2}}{2} \tag{4.27}$$

where $\theta_{n,i}$ $\theta_{n+1,i}$ are joint coordinates of consecutive particles n and $n+1$, defined according to the *particle-oriented approach.*

Assuming the current system configuration ($\boldsymbol{\theta}$) and the actuation force $T = T_0$ to be known, the forces $\mathbf{f}_n$ and $\mathbf{f}_n^\top$ can be derived from Eq. 4.26 and Eq. 4.27. In particular, the former equation is applied recursively for each particle in the chain. The forces $\mathbf{f}_n$ and $\mathbf{f}_n^\top$ act on the link Ω_n. The resultant of these forces is added to the force vector $\mathbf{F}^e$ of Eq. 4.25 to compute the effect of the external forces on the system. In particular, the resultant torque $\tau_n = \frac{e}{2}(\mathbf{f}_n + \mathbf{f}_n^\top)$ is derived taking into account the distance e between the eyelets of an Ω-type link, as shown in Fig. 4.7.

III. **Unlocking Forces.** The forces that the built-in Shape-Memory Alloy (SMA) actuators exert to unlock a folding side of a particle (see Sec. 3.4.1), are modeled as external forces acting on the joints $\mathbf{v}_1$ and $\mathbf{v}_1^{(D)}$ (Fig. 4.8b). Such forces are added to the corresponding elements of the external force vector $\mathbf{F}^e$. As the latter indicates linear forces and torques applied to the center of mass of each link composing a particle chain, the linear force that an SMA actuator exerts on the δ-type link i is transformed into linear forces and torques with respect to its center of mass. In particular, letting $\mathbf{f}$ be the force that the SMA actuator exerts on link i, the torque τ_i applied to the center of mass of link i is $\tau_i = \frac{d}{2}\mathbf{f}$, where d is the length of a δ-type link.

Internal Forces $\mathbf{F}^i$

Internal forces include all the reaction forces at the joints which are due to the kinematic constraints introduced in Sec. 4.3.3. As anticipated, three types of constraint exist that cause reaction forces: *loop*, *joint-limit* and *rotational-friction* constrains. In the following, we analyze the internal forces relative to each constraint.

i. **Loop Constraint Forces.**

The kinematic equations Eq. 4.14 and Eq. 4.15 are derived for the unconstrained system, where the kinematic loops are temporarily removed. The loop constraint equation $\Phi^c(\boldsymbol{\theta}) = \mathbf{0}$ (Eq. 4.11) defines an implicit condition for the joint coordinates $\boldsymbol{\theta}$ to restore the loop closure. Referring to Fig. 4.8b, the reaction forces necessary to satisfy the loop constraint, namely to ensure that $\mathbf{v}_3$ and $\mathbf{v}_2^{(D)}$ coincide, act between the two branches having $\mathbf{v}_3$ and $\mathbf{v}_2^{(D)}$ as extremes and originate from the physical contact between the joints $\mathbf{v}_3$ and $\mathbf{v}_2^{(D)}$.

We can derive these reaction forces from Eq. 4.25 and the Jacobian $\boldsymbol{J}^c = \frac{\partial \Phi^c(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$ defined in Eq. 4.13. In order to have the constraint equation $\Phi^c(\boldsymbol{\theta}_k) = \mathbf{0}$ satisfied, also its derivative $\boldsymbol{J}^c \cdot \dot{\boldsymbol{\theta}}_{\boldsymbol{k}} = \mathbf{0}$ needs to be null. Thus, by multiplying both sides of Eq. 4.25 by $\boldsymbol{J}^c$, we can derive $\mathbf{F}^i$ from the following equation, which also needs to be null:

$$\boldsymbol{J}^c\dot{\theta}_k = \boldsymbol{J}^c(\boldsymbol{B}^\top \boldsymbol{M}\boldsymbol{B})^{-1}\boldsymbol{B}^\top(\mathbf{F}^e + \mathbf{F}^i)\cdot\Delta t + \boldsymbol{J}^c\dot{\theta}_{k-1} \equiv \mathbf{0} \tag{4.28}$$

The reaction forces between $\mathbf{v}_3$ and $\mathbf{v}_2^{(D)}$ can be expressed in form of Lagrange's multipliers $(\boldsymbol{J}^c)^\top \cdot \lambda$. We recall from Eq. 4.12 that $\boldsymbol{J}^c \cdot \dot{\boldsymbol{\theta}} = \frac{\partial \mathbf{v}_3}{\partial \boldsymbol{\theta}} \cdot \dot{\boldsymbol{\theta}} + \frac{\partial \mathbf{v}_2^{(D)}}{\partial \boldsymbol{\theta}} \cdot \dot{\boldsymbol{\theta}}$, which means that each column of $\boldsymbol{J}^c$ indicates an admissible moving *direction* of the points $\mathbf{v}_3$ and $\mathbf{v}_2^{(D)}$ due to a variation of the relative joint coordinates $\boldsymbol{\theta}$. As the reaction forces can only produce motion along such admissible directions, the reaction forces are a linear combination of the columns of $\boldsymbol{J}^c$ represented by the product $(\boldsymbol{J}^c)^\top \cdot \lambda$. The latter yields the torques τ_i acting at each joint i with $i \in [1..5]$. It is possible to demonstrate that the relationship between internal forces $\boldsymbol{F}^i$ (expressed in absolute coordinates) and the corresponding torques at the joints is $\boldsymbol{B} \cdot \boldsymbol{F}^i = \tau$, with $\boldsymbol{B}$ being the matrix defined in Eq. 4.15.

By substituting $\boldsymbol{B} \cdot \boldsymbol{F}^i$ in Eq. 4.25 with $\boldsymbol{B} \cdot \boldsymbol{F}^i = (\boldsymbol{J}^c)^\top \cdot \lambda$, the multiplier λ is derived by solving the following linear system:

$$\underbrace{\boldsymbol{J}^c(\boldsymbol{B}^\top \boldsymbol{M} \boldsymbol{B})^{-1}(\boldsymbol{J}^c)^\top}_{\boldsymbol{A}} \cdot \lambda = \underbrace{-\boldsymbol{J}^c(\boldsymbol{B}^\top \boldsymbol{M} \boldsymbol{B})^{-1}\boldsymbol{B}^\top \mathbf{F}^e - \boldsymbol{J}^c \frac{\dot{\theta}_{k-1}}{\Delta t}}_{\mathbf{b}} \tag{4.29}$$

ii. **Joint Limit Forces.** Following the same reasoning applied to derive the loop constraint forces, we obtain the joint limit force in form of Lagrange's multipliers $(\boldsymbol{J}^u)^\top \cdot \lambda$, where $\boldsymbol{J}^u$ is the Jacobian defined in Eq. 4.21. In order to satisfy loop and joint limit constraints at the same time, Eq. 4.29 need to be reformulated as follows, where $\boldsymbol{J}^c$ and $\boldsymbol{J}^u$ are concatenated into the Jacobian $\boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}^c \\ \boldsymbol{J}^u \end{bmatrix}$:

$$\underbrace{\boldsymbol{J}(\boldsymbol{B}^\top \boldsymbol{M} \boldsymbol{B})^{-1}\boldsymbol{J}^\top}_{\boldsymbol{A}} \cdot \lambda = \underbrace{-\boldsymbol{J}(\boldsymbol{B}^\top \boldsymbol{M} \boldsymbol{B})^{-1}\boldsymbol{B}^\top \mathbf{F}^e - \boldsymbol{J} \frac{\dot{\theta}_{k-1}}{\Delta t}}_{\mathbf{b}} \tag{4.30}$$

iii. **Rotational Friction Forces.** The dynamic friction at the joint i generates a torque $\widetilde{\tau}_i$ that counteracts the angular velocity θ_i. We model this condition by introducing the following equation:

$$\widetilde{\tau}_i = -\mu_d \theta_i \tag{4.31}$$

where μ_d is a dynamic coefficient of friction.

As $\widetilde{\tau}_i$ does not generate any linear force, we add each $\widetilde{\tau}_i$ to the corresponding torque τ_i^i of vector $\mathbf{F}^i = [\cdots \ f_{x_i}^i \ f_{y_i}^i \ \tau_i^i \ \cdots]^\top$.

4.3.7 Evaluation

We compare the two dynamic model formulations (chain-oriented and particle-oriented) and evaluate their accuracy to predict the dynamic behavior of an actuated particle chain. The particle-oriented dynamic model is derived in the previous section, while for the chain-oriented dynamic model we refer to Paper C.

We actuate a prototype with a predefined sequence of actions and provide the same sequence as input to the models under evaluation. The latter predict the state of the chain at each point in time. To estimate the divergence of the dynamic models' predictions from

reality, we apply the same actuation input to the prototype and the models and compare the status of the chain at corresponding points in time. In particular, we consider absolute position and inclination of the bottom Ω-type link of the particle chain.

4.3.8 Experimental Setup

A three-particle prototype executes the folding experiment reported in Table 4.1, where particles are sequentially unlocked and folded starting from the bottom particle and proceeding upwards until all the particles are completely folded. At predefined points in time, the built-in actuator of a selected particle (i.e., SMA) unlocks one of the folding sides. The local actuator remains active for a constant interval of about 0.69 s. While the SMA actuator is still active, the external actuator is activated to fold the chain. Table 4.1 reports for each particle the folding **side**, the **start** and the **end** time of the unlocking and folding phases.

With the aid of a video camera, we record the prototype executing the above experiment. After having corrected the inevitable distortion due to the misalignment of camera and prototype, we analyze the video and annotate for each point in time position (x, y) and angular displacement (α) of the bottom link of the chain. We consider this link to be the best representative of the status of the entire chain.

The two dynamic models derived according to Chain-Oriented Approach (COA) and Particle-Oriented Approach (POA) are applied to predict the behavior of the chain undergoing the folding experiment reported in Table 4.1. We compare the obtained predictions against the behavior of the prototype. The synchronization between the prototype and the output of the models is calculated on the basis of the frame rate of the video.

4.3.9 Comparison

The diagrams of Fig. 4.11 show for each point in time the accuracy of the prediction of two dynamic models compared with the behavior of the prototype. The upper diagram shows the inclination α_n of the bottom link of the chain, while the lower diagram reports the coordinates x_n and y_n of the center of mass. The vertical dashed and solid lines indicate, alternately, the activation time of the particle unlocking mechanism and of the external actuators, as reported in Table 4.1.

We observe that until the second particle is unlocked (time = 3.2 s), the behavior predicted by both models matches the behavior of the real prototype with average absolute errors (for COA) lower than $0.049\,rad$, $1.15\,mm$, $1.80\,mm$ respectively for α_n, x_n and y_n.

Table 4.1: Folding Experiment

Particle	**Unlock Particle**			**External Actuator**	
	Side	**Start** [s]	**End** [s]	**Start** [s]	**End** [s]
3. (bottom)	Right	1.45	2.14	1.94	3.00
2.	Right	3.29	3.98	3.79	4.91
1. (top)	Left	5.133	5.82	5.63	6.66

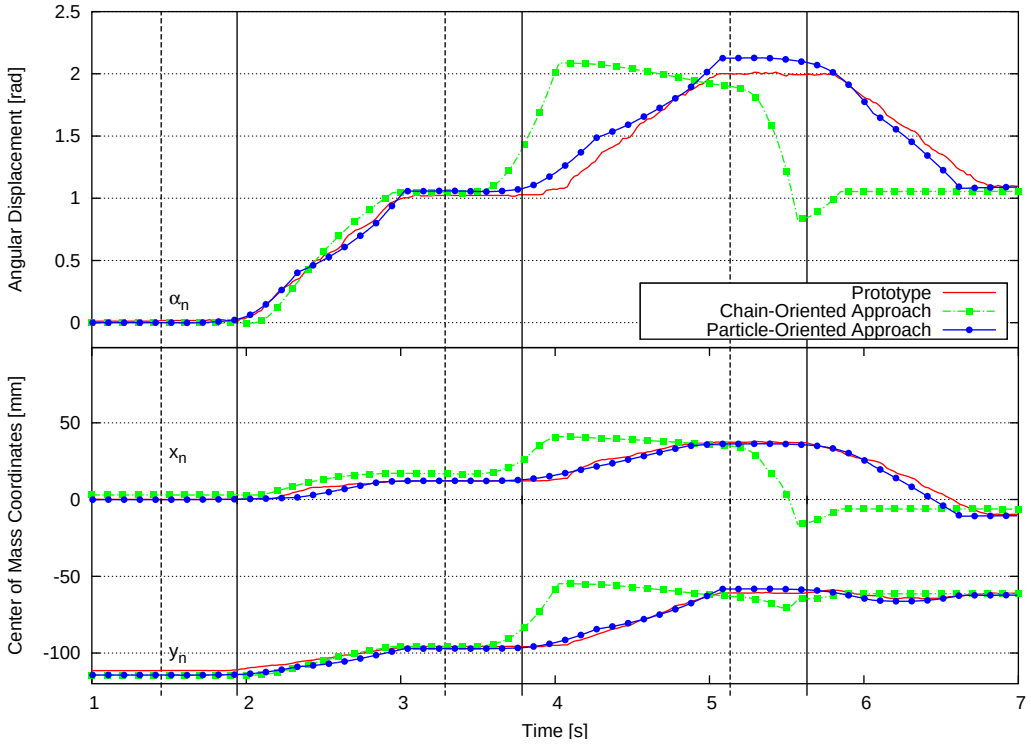


Figure 4.11: Evaluation of the dynamic models against reality.

About 0.1 s after unlocking of the second particle, the prediction based on COA starts diverging quite significantly from the real prototype, whereas the prediction based on the POA follows the behavior of the real system with more accuracy. Nonetheless, in both cases the final postures predicted by POA and COA have maximum relative errors lower than 0.035, 0.36 and 0.018 on the coordinates (α_n, x_n, y_n) at time 7 sec. The best results can be achieved with POA whose relative errors decrease to 0.004, 0.082, and 0.018 on the coordinates (α_n, x_n, y_n) at time 7 sec.

However, also the COA model can eventually predict whether actuation forces are sufficiently strong for a particle chain to reach the final target configuration. The discrepancy that we observe when applying the COA model is mainly a temporal offset in the reaction time of the compared systems. The COA prediction anticipates the behavior of the real system of about 0.3 s. As explained in Paper $\boxed{C}$, we conjecture that the reason behind this divergence lies in the empirically derived set of actuation forces applied as input to the model. An actuation force of $3.0\,N$ is initially set to simulate the action of the external actuator and exponentially decreased to a value of $0.5\,N$, which is the minimum force necessary to retain the posture of the chain after each actuation, whereas any lower value makes the chain unfold. The combined effects of this constant retention force and the unlocking force applied to the folding edge of the particle, result in an early folding of the chain.

More consistent results derive from the application of the POA model. In this case,

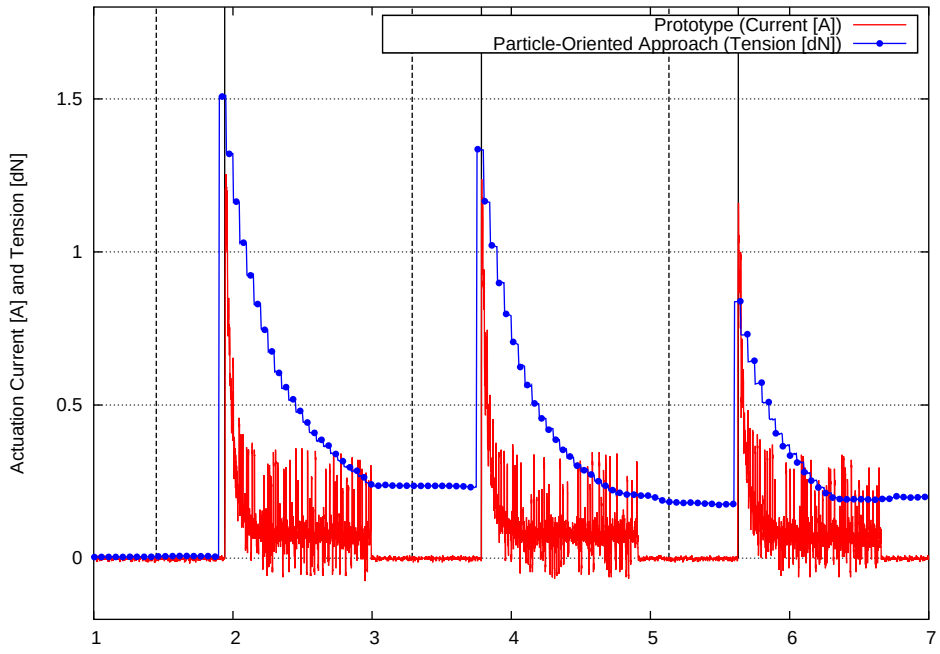


Figure 4.12: Comparison of the predicted actuation forces and the electric current applied to the DC motor of the prototype, which is proportional to the torque that the motor can produce.

a set of minimal actuation forces, input to the model, is computed to obtain a constant winding velocity of the tendons and to retain the posture of the chain when the external actuator is inactive. An explanation about the computation of this set of forces is provided in the following section.

4.3.10 Model-Predictive Control

One motivation behind the derivation of a dynamic model is its application in a model-predictive controller to infer a set of minimal actuation forces necessary to actuate the system and thereby prevent damages to the chain. To demonstrate the feasibility of this approach, we consider the actuation forces that the external actuator needs to exert in order to obtain the dynamics shown in the diagram of Fig. 4.11. In particular, we notice that the motor of the prototype winds up the tendons at a constant velocity. To obtain a similar behavior, we apply the POA dynamic model to compute a set of minimal forces to ensure a constant winding velocity of the tendons tied to the differential winches.

From the measurements executed on the existing prototype, and reported in Fig. 4.11, we derive the average tendon winding velocity. Although the derived dynamic model does not allow us to directly infer the actuation forces given a target velocity (for which an

inverse dynamic model would be required), we apply heuristic techniques to find optimal forces to actuate the system, as introduced in Sec. 4.1.2. To efficiently apply this approach, we adjust the intensity of the actuation forces proportionally to the difference between the target and the current winding velocity. Fig. 4.12 shows the obtained actuation force that, for each actuation, starts from a maximum value and exponentially decreases to a minimum constant value necessary to retain the folded chain.

To demonstrate the consistency of the model with reality, we compare the obtained result with the actuation force applied to the prototype. As the latter is not easily measurable we consider the input current applied to the DC motor, which is proportional to the output torque of the motor [123], in turn proportional to the tension applied to the tendons. As Fig. 4.12 shows, the tension inferred through application of the POA model has an exponential trend similar to the current applied to the DC motor actuating the prototype, as long as the DC motor is active. When the motor is not active, the input current is set to zero, while the internal friction of the motor reduction gearbox is sufficient to prevent the tendon from unwinding and thus the chain from unfolding. In other words, despite the current of the motor being null, a tension still persists on the tendons of the prototype to retain the final posture of the chain. As we observe in Fig. 4.12, a minimal retain tension is also predicted through the application of the dynamic model.

We can conclude that the derived dynamic model is consistent with reality, considering that, for a similar input applied to both the prototype and the POA dynamic model (Fig. 4.12), the latter is able to predict the behavior of the prototype with acceptable accuracy (as Fig. 4.11 demonstrates). In addition, the derived dynamic model can be applied to infer optimal actuation forces that minimize the tension on the chain and maximize the scalability of the whole system.

4.3.11 Limitations

Although the kinematic and dynamic models derived in this section overcome most of the limitations affecting the static model (Sec. 4.2.6), the following limitations exist:

1. The dynamic model does not provide any explicit information about the reaction forces applied to the joints. However, it is possible to infer these forces through the method of the Lagrange's multipliers, in the same way that the loop constraint reaction forces are derived (page 90).

2. Limitation 3 of the static model persists: it is not possible to estimate the mechanical stress on the components of the chain. This would require finite element methods, out of the scope of this work. In any case, this limitation does not compromise the application of the dynamic model in optimal planning and control. However, future work should address this issue for a more complete and reliable representation of the whole system.

Chapter 5

Conclusion and Future Work

This chapter summarizes the contributions and proposes potential future work and research in the covered topics.

5.1 Contribution

This doctoral thesis addresses the challenges of designing and coordinating an ensemble of modular robotic particles for the realization of shape-shifting materials. In particular, we focus on (i) mechatronic solutions that allow the formation of arbitrary shapes with high-resolution rendering, and on (ii) planning and control approaches to coordinate the ensemble of modular robotic particles performing a shape shift.

While on the one hand, the formation of arbitrary shapes requires particles to freely rearrange, on the other hand, the sophisticated mechanisms which are necessary to make this possible hamper particle miniaturization and system scalability. Aiming at high-resolution rendering, we demonstrate that particle miniaturization and system scalability can nevertheless be achieved by trading particle mobility for a simpler mechatronic design.

Towards this goal, we design and build prototypes consisting of modular robotic particles concatenated into a chain and remotely actuated by means of an external actuator. While particles cannot modify their topology, they can control the local curvature of the chain by folding. In this way, a single chain can outline arbitrary planar shapes, and multiple chains can be arranged to form a shape-shifting surface. This leads to the concept of 3D shape-shifting display, where each chain outlines the contour of a slice of a target 3D object to be displayed. It follows that a reduced particle mobility does not necessarily compromise the formation of arbitrarily complex shapes.

Additionally, we devise the "Force-Guiding Principle" whereby particles exploit an externally provided force to actuate, instead of relying on a locally generated force. Essentially, as the strength and size of actuators are typically correlated, the presence of actuators embedded in particles limits the scalability of the system to the maximum force actuators can exert or calls for stronger actuators, hence, bigger particles. The "Force-Guiding Principle" overcomes this problem by averting particle's local actuators. This improves system scalability and facilitates particle miniaturization for higher resolution rendering.

We also observe that appropriate particle coordination is essential to maximize the

scalability of the system performing a shape shift. Intense actuation forces might indeed arise in case of inadequate coordination, with undesired consequences for the mechanical stability and integrity of the system. To prevent such a situation, optimal planning and control are required to find shape shifting solutions that minimize the maximum actuation forces and thereby maximize the number of particles that the system can cope with. To this end, we derive static and dynamic models to predict the behavior of the system subject to actuation forces. The predictions indicate whether a target shape can be achieved through the application of maximum admissible actuation forces. Furthermore, we show that the dynamic models can be used in a model-predictive controller to infer a set of minimal actuation forces necessary to actuate the physical system.

5.2 Future Work

In the broader vision of programmable matter and shape-shifting material, plenty of work still needs to be carried out. In this section, we keep focussing on the realization of a Shape-Shifting Display (SSD) and discuss some limitations of the present work, which can be considered for future work.

5.2.1 Mechatronic Design

In this work, we demonstrate the feasibility of the "Force-Guiding Principle" to improve the scalability of a particle chain and to facilitate particle miniaturization for high-resolution rendering. Our assumption is that a simpler mechatronic design enables a higher resolution rendering. To prove this concept, advanced mechatronic solutions need to be explored in order to scale down the current physical dimensions of a particle by at least two orders of magnitude, namely reaching the tenth of millimeter scale. This might imply the use of Micro-Electro-Mechanical Systems (MEMS) technology in combination with electroactive polymers for actuation.

5.2.2 Manufacturing Process

Future improvements to the mechatronic design should also address challenges concerning the manufacturing process. Although in this thesis we have not focussed on this aspect (we refer to paper $\boxed{A}$ for further details), the assembly of a particle chain requires a long patient handwork. While the links forming a particle are designed with snap-fit joints to facilitate their assembly, the manual insertion of the Shape-Memory Alloy (SMA) wires into the narrow cavity obtained in the rigid links (which is necessary to extend the length of the SMA wire and thus its shrinking effect) is a critical and time consuming operation. Difficult is also interconnecting the SMA wires to the electronic board. Because of its chemical composition, SMA cannot be directly soldered to other metallic components and also chemically altering the SMA composition is not a valid solution, according to SMA manufactures. Similarly, the use of commercial wire terminals or other crimping solutions to fix the SMA to the board is not practicable in our case, due to the small dimensions of the electronic board. In order to ensure both electrical conductivity and mechanical stability, our workaround is to wedge the SMA into the hole of the circuit board with a short piece of copper wire that is eventually soldered to the board.

While the manual manufacturing approach can be suitable for producing a few tens of participles, it becomes infeasible for a larger number of particles especially if their dimensions range in the sub-millimetric scale. Future work should consider automatic manufacturing processes to overcome this issue.

5.2.3 Communication Protocol

The prototype presented in this work relies on the 1-Wire™ technology to establish a master-slave communication between the main controller and each particle. Aiming at high system scalability, this solution poses remarkable limitations considering that the 1-Wire™ bus does not allow simultaneous communication among slaves nor master-slave role exchange to support particle-to-particle communication. This means that the local interaction between neighboring particles needs to be coordinated by the main controller and cannot be locally handled. Even assuming that each chain in an SSD has a dedicated 1-Wire™ bus this limitation still persists.

Despite not reported in this thesis, to overcome the limitations related to the 1-Wire protocol, in [96] we explore the feasibility of a point-to-point communication protocol, which exploits the pair of SMA wires embedded in each particle as a medium to support full-duplex serial communication. This aims at increasing the scalability of the system by supporting local communication among particles and consequently enabling communication to take place simultaneously among multiple pairs of particles. A major challenge to be faced with point-to-point communication protocols concerns the synchronization among particles and the external actuator, as the activation of the external actuator can only occur within a short time interval after particles have unlocked a folding side. In [96] a solution to ensure synchronization among nodes is proposed, which provides an accuracy of 10 ms over 12 nodes. Future work is required to improve stability and accuracy of the synchronization algorithms.

Another limitation of the 1-Wire™ protocol lies in the addressing mode. Each node forming a 1-Wire™ network has a unique factory hard-coded identifier, used as network address. As these identifiers are unknown until the device is networked, a discovery algorithm is required to identify the nodes in the system (see Paper $\boxed{A}$). In addition, localization algorithms are required to infer the position of each particle in the SSD and map it to the corresponding network address. This is a fundamental requirement to properly configure particles and obtain the target shape to be displayed. To overcome these limitations, in [96] we propose a spatial addressing mode as part of the point-to-point communication protocol mentioned above, whereby the main controller establishes a communication link with a particle or group of particles of an SSD based on their physical location in the system. Future work on the addressing mode should further explore this approach and evaluate its feasibility.

Visible light communication can provide a valuable solution to configure all the particles of an SSD at once. This could substitute wired protocols, or be complementary to the latter. A possible idea is that all the particles of an SSD are simultaneously programmed by means of flashlight transmitting the final configuration to each particle. This approach still requires synchronization between the external actuator and the particles, which can also rely on a flashlight or a wired communication protocol. Future work should explore this aspect that can become of interest for the creation of Shape-Shifting Materials (SSMs)

and Programmable Matter (PM).

5.2.4 Optimal Planning

In this thesis, we considered the importance of optimal planning for ensuring scalability, as demonstrated in Chap. 3 and Chap. 4, even though this aspect has not been exhaustively investigated. In particular, the problem of efficiently exploring the large state space of a particle chain, in order to find a sequence of intermediate configurations (see Sec. 4.1.1), needs to be solved. Towards this goal, the static and dynamic models presented in this thesis can be applied as proposed in Chap. 4 to support future work.

5.2.5 Optimal Control

In combination with optimal planning, optimal control is required to infer a set of minimal actuation forces to compensate for the absence of sensors in particles. To this end, we propose in this thesis a heuristic method to estimate these forces through the application of the dynamic model derived in Chap. 4. This approach, however, does not provide an exhaustive solution to the optimal control problem. Future work is required to investigate this aspect and identify efficient solutions to control the transition between intermediate configurations.

5.2.6 Model Extension

The dynamic model derived in Chap. 4 predicts the behavior of a particle chain subject to actuation forces with acceptable accuracy. The proposed model takes into account the mechanical constraints among the rigid links of a particle chain, in particular loop and joint limit constraints, and the forces due to the built-in and external actuators. One aspect that the current model does not encompass is the relationship between the overall length of the tendon deployed along the chain and the length of its path through the traversed eyelets. As the path of the tendon depends on the spatial arrangement of the eyelets, the sum of the distances between consecutive eyelets should match the overall length of the tendon. This defines an additional kinematic constraint that should be considered in future work in order to obtain a more realistic and complete model.

5.2.7 User Interaction

In this doctoral thesis we have mainly focused on design aspects that allow us to obtain a high resolution rendering SSM, for which we have proposed solutions from both the hardware and software perspectives. In particular, we have demonstrated that high system scalability and particle miniaturization can be obtained with simple mechatronic design. Future work should quest whether simple mechatronic design can also allow advanced human-matter interaction paradigms, whereby commands to control the shape and the features of SSM are encoded in hand-touch and gestures. For example, through hand-touch users can “mold” the shape of an SSM as if it were made of clay, while with gestures users can mimic the movement of a paint brush to modify the surface color.

Publications

This Thesis is based on the following peer-reviewed conference papers (sorted by publication date):

[A] Matteo Lasagni and Kay Römer. Force-guiding particle chains for shape-shifting displays. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago (IL) USA – September 2014. DOI: 10.1109/IROS.2014.6943112

[B] Matteo Lasagni and Kay Römer. Force model of a robotic particle chain for 3d displays. *The 30th ACM/SIGAPP Symposium On Applied Computing (SAC) – track: Intelligent Robotics and Multi-Agent Systems (IRMAS)*. Salamanca, Spain – April 2015. DOI: 10.1145/2695664.2695932

[C] Matteo Lasagni and Kay Römer. Programmable Robotic Chains: Kinematics and Dynamics of a Scalable Tendon-Driven Under-Actuated Multibody System. *IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. Banff, Canada – July 2016. DOI: 10.1109/AIM.2016.7576993

[D] Matteo Lasagni and Kay Römer. Dynamic Model of Tendon-Driven Roboitc Chains Forming a Shape-Shifting Surface. *ASME Conference on Smart Materials, Adaptive Structures and Intelligent Systems (SMASIS)*. Stowe (VT) USA – September 2016. DOI: 10.1115/SMASIS2016-9263

Paper A

DOI: 10.1109/IROS.2014.6943112
Link: http://ieeexplore.ieee.org/document/6943112/
ISBN: 978-1-4799-6934-0

Force-Guiding Particle Chains for Shape-Shifting Displays

Matteo Lasagni and Kay Römer

***Abstract*— We present design and implementation of a chain of particles that can be programmed to fold the chain into a given curve. The particles guide an external force to fold, therefore the particles are simple and amenable for miniaturization. A chain can consist of a large number of such particles. Using multiple of these chains, a *shape-shifting display* can be constructed that folds its initially flat surface to approximate a given 3D shape that can be touched and modified by users, for example, enabling architects to interactively view, touch, and modify a 3D model of a building.**

I. Introduction

The underlying goal of this work is the design and implementation of a "shape display" – a two-dimensional surface that can fold into the third dimension where the shape to be displayed can be freely programmed and dynamically changed. A user can not only view and touch the displayed 3D surface, but by means of touch gestures recognized through sensors built into the surface could interactively modify the displayed shape. Such a shape display would enable a wide range of interesting applications, for example, an architect could display and interactively modify a newly designed building; scientists could display complex 3D graphs and models to better understand them.

One approach to realize such a shape display is based on many tiny modular robots (i.e., *particles*) that can change their arrangement such that the aggregate surface formed by all the robots forms the displayed shape. However, approaches based on freely moving autonomous particles have the disadvantage that each particle needs complex actuators and latches to move into a desired configuration and latch into a mechanically stable shape. Also, power supply and communication among particles is challenging as they form dynamically changing connection topologies. Therefore, researchers have investigated approaches where neighboring particles are connected by joints and can change their relative orientation by means of actuators. The resulting fixed connection topology allows for wired networking and power supply, but actuators and latches are still needed in each particle. Specifically, these actuators and latches need to be strong enough to create large and mechanically stable particle aggregations. This represents a hurdle towards miniaturization of the particles, effectively limiting the "resolution" of the shape display to rather coarse structures.

Our contribution lies in removing the need for latches, only requiring mechanically weak and simple actuators in each particle, therefore enabling a better miniaturization of the particles. We call our approach "force-guiding particles", as a single strong actuator exerts a mechanical force on all particles in a chain and a weak and simple actuator inside each particle guides this force to fold the chain into a desired configuration. As the particles in the chain are connected by joints, two wires running through the chain provide power to and establish a communication network among all particles. Therefore, miniaturization of particles becomes feasible. By placing multiple such chains (all driven by a single external actuator) next to each other we obtain a "shape-shifting curtain" that can display 3D surfaces.

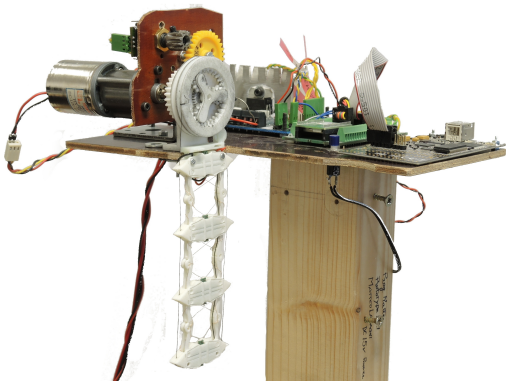


Fig. 1: Our prototype of a "force-guiding particle" chain.

The remainder of the paper describes the electro-mechanical design of a force-guiding particle chain, as well as algorithms for control and planning of chain folding to approximate a desired shape. We describe a prototype (Fig.1) that has been produced with a 3D printer and characterize its performance. We also provide an analysis of the scaling properties of force-guiding particle chains.

II. Related Work

"Shape-Shifting Material" and "Programmable Matter" are general terms for research on materials whose physical properties and in particular shape can be changed dynamically under program control or through direct interaction [1], [2], [3]. The approach is typically based on an extreme form of modular robotics, where small robotic "particles" can change their relative position by means of suitable actuators.

One can broadly identify two sub-classes of approaches. In the first the robots are detachable and can "climb" each other (e.g., [4], [5], [6], [7], [8], [9]). While arbitary shapes can be formed, the individual robots are typically complex

Matteo Lasagni and Kay Römer are with the Institute for Technical Informatics, Graz University of Technology, 8010 Graz, Austria {lasagni,roemer}@tugraz.at

as they need complex actuators and latches as well as sophisticated power supply and networking as the robots are not permanently connected. That leads to costly robots which cannot be easily miniaturized. Even when relocation relies on non-moving parts (e.g., magnets [5], [10]), miniaturization might still be difficult as each particle needs to excert strong forces on neighboring particles in order to form mechanically stable shapes consisting in a large number of particles.

In the second sub-class "robots" are connected in fixed topologies and the resulting substrate can be deformed by embedded actuators. These substrates can be two-dimensional and fold along crease patterns inspired by Origami (e.g., [11], [12], [13]), or one-dimensional chains that fold in 2D (e.g., [14]) or 3D (e.g., [15], [16]). While this reduces the complexity of the individual robots due to the fixed connection topology, miniaturization of the individual robots is still difficult as the force to fold and latch is still created by actuators inside each robot.

To overcome this intrinsic limitation, White et al. [17] devise an external manipulator to fold a chain of passive latching particles into 3D shapes. However, their external actuator is rather complex (i.e., essentially a robotic arm) and a separate actuator is needed for each chain, thus it does not scale well to larger systems with multiple chains. Also, the unfolding of the chain requires manual support. Hence, this approach is not suitable for a "shape display" as we envision it where displayed shapes can be dynamically changed.

III. Requirements

Below we explicate the main requirements on force-guiding particle chains.

a) Formation of arbitrary shapes: It should be possible to display a wide range of connected 3D surfaces. As our shape display consists of multiple chains placed next to each other, each chain should be able to approximate a 2D slice (i.e., curve) of the 3D surface.

b) Miniaturization of particles: To enable a high-resolution display, it should be possible to miniaturize particles. This requires particles to have a simple structure. In particular, it should be possible to miniaturize the mechanical structures and actuators contained in a particle without compromising the mechanical stability of the particle chain.

c) Scalability: As particles are miniaturized, a growing number of particles are required to display surface shapes of realistic size. Therefore, it should be possible to include a large number of particles in each chain and to include many such chains into a shape display, i.e., it should be possible to scale up the number of particles. Again, mechanical structures and actuators play a key role here as they need to be strong enough to deal with the growing extension and weight of the chain as the number of particles incrases.

IV. Approach: Force-guiding Particles

Our shape display consists of many particle chains placed next to each other, thus forming a 2D surface. Each chain can fold within a plane that is perpendicular to this surface. Together, all folded chains approximate the 3D shape that is to be displayed.

Each chain resembles an articulated ladder-like skeleton (Fig. 2(a)). Here, a particle consists of the square formed by two consecutive rungs and the two edges connecting these rungs. By folding one of the edges, the square can be reconfigured into a triangle, thus deflecting the chain. As each particle in the chain can be folded left or right, the chain can approximate any planar curve that does not self-intersect. Together, all chains can thus approximate any connected 3D surface, thus meeting the requirement of displaying arbitrary shapes. We outline a planning algorithm to approximate a given shape later in the paper.

The regular geometry of the structure allows a strong external actuator to exert a compressing force F_w on the whole chain by pulling two threads that run through the ladder structure on the left and on the right as shown in Fig. 2(a). By exerting a small force F_s on one of the edges, the latter slightly bends, such that F_w will fully compress the edge, thus transforming the rectangular particle into a triangle. Edges to which no force F_s is applied remain locked in straight configuration even if F_w is applied. If F_w is released, the chain returns into straight configuration due to gravity. Thus, particles "guide" the external force for folding.

The single external actuator (i.e., motor) is dimensioned such that it can pull the threads of *all* chains even if the number of particles in each chain becomes large. While that may imply a relatively big actuator, the dimensions of the particles (and thus the resolution of the shape display) are not affected. In that sense, we can scale up the number of particles in a chain without impact on the size and complexity of the particles, thus meeting the scalability requirement.

As we will explain in the following section, the force F_s in the particle can be generated by means of a Shape Memory Alloy wire that contracts when applying a current. By retaining the external force F_w, the chain remains folded without the need for latches inside each particle. Thus, the mechanics of a particle essentially consists of hinges, joints, and SMA wires – making it very simple and enabling miniaturization, thus meeting the miniaturization requirement.

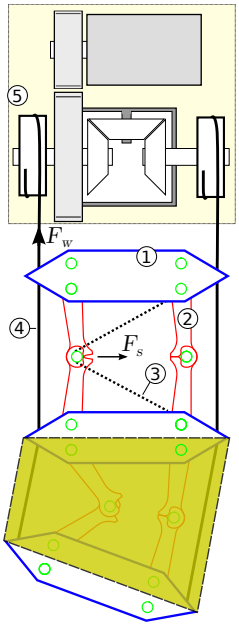
V. Hardware and Software Design

In this section we describe in more detail the design of force-guiding particle chains. We begin with a description of the mechanics and electronics, followed by discovery and localization of particles, and conclude with control and planning aspects.

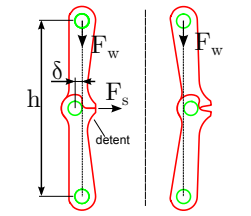
A. Mechanical Design

Figure 2(a) shows the design of a particle chain, where a deformable particle correspond to the highlighted area. A particle chain consists of the following five main components (see also the number labels in the figure).

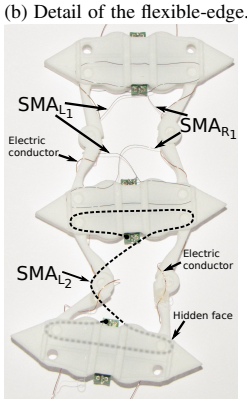
1. Rigid edge. Each particle consists of two rigid horizontal edges that are shared with the particles above and below. The particular stretched-hexagon shape ensures a precise



(a) Mechanical Design.



(b) Detail of the flexible-edge.



(c) SMA path extension.

Fig. 2: Mechanical components of the system.

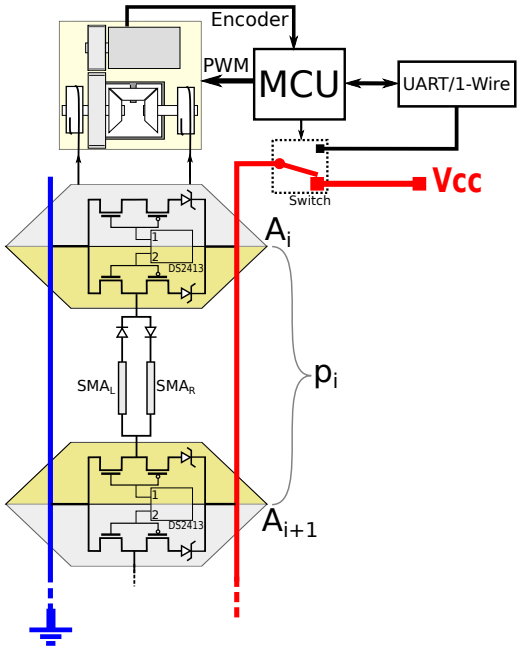


Fig. 3: Electrical block diagram.

triangular mesh when the particles fold. Its dimensions are tailored to encase the electronics that control the particle.

2. Flexible edge. Hinged to successive rigid edges via snap-fit joints, each of the two flexible edges of a particle embodies a monostable mechanism to lock/unlock the folding. As shown in Fig. 2(b), the slight misalignment of the middle hinge w.r.t. the two hinges at the extremes, combined with a counteracting detent, lead to a "locked" straight configuration even when the external force F_w is applied. When, instead, a small force F_s pulls the middle hinge towards the centre of the particle, the link gets "unlocked", and F_w deforms the particle into a triangular configuration.

3. Active strap. Made of Shape Memory Alloy (SMA), the contraction due to an electric current heating it up causes the lateral force F_s to unlock the flexible edge as described above. When the current is stopped, the SMA cools down and returns to the original length. Since the SMA features a relative contraction of about 4%, a longer SMA wire running through the opposite rigid edges amplifies the contraction effect by a factor of four as shown in Fig. 2(c).

4. Actuation thread. Next to the flexible edge, the actuation thread extends along the chain to propagate the force F_w the external actuator provides. The resulting effect on the chain is a compressing force on each side that, combined with the unlocking of a flexible edge, causes the deflection of the chain. By applying a constant tension, the chain retains the folded configuration without a need for latches.

5. External actuator. It provides the mechanical power to the system. An electric motor rolls up the two actuation threads, by means of spoolers connected via a differential gear. Because of the misalignment between flexible edges and threads, a particle deformation makes one thread roll up, while the other one slightly unrolls. To mechanically compensate this effect, the differential gearbox decouples the two spoolers. In this way, a single motor can even drive multiple chains connected to the same shaft, provided an equivalent number of deforming particles among chains.

B. Electronics

Figure 3 shows the electronic architecture of the system, which consists of an MCU that controls the external actuator as well as the particles via a 1-wire bus that also supplies power to the particles.

MCU. Core of the system is an 8-bit MCU (ATmega128RFA1) that controls all the components: it remotely drives the local actuation of each particle and synchronizes the external actuator to fold the chain. As all the particles share the same bus to communicate with the MCU and power the SMA straps, the MCU controls the bus line switching it from communication mode (1-Wire®) to SMA power supply mode (Vcc) and vice-versa.

Actuation and feedback loop. The main actuator based on a 12 V DC motor (HN-34PGD-2416T) can rotate at variable speed (PWM controlled) in both directions to fold and unfold the chain. The planetary gear reduction embedded in the motor (gear ratio 410:1) prevents the backward rotation

so the motor can be switched off once the chain has been folded. As no sensors are embedded in the particles, a quadrature encoder (48 cycle/rev.) connected through a gear (ratio 1:3) to the main shaft, allows feedback control to stop the motor when a desired configuration has been reached.

1-Wire® bus. Two conductors conveniently threaded through the chain form the data line and ground reference for a 1-Wire® bus that connects particles and MCU. 1-Wire® uses a master/slave protocol and supports data and power on the same bus. Power is supplied either when no communication is in progress or when a logic '1' is transmitted. In these two cases, a pull-up resistor (in the master) pulls the voltage on the bus to 5 V. While the logic '1' passively results from the pull-up resistor, the dominant '0' is obtained by short-cutting data line and ground. This limits the maximum current on the bus to 5.4 mA, as higher currents can introduce logic '0' in the communication. Each 1-Wire® slave exploits the period when the bus is powered to harvest energy in a small capacitor. As interface between the MCU and the bus, we adopted a UART to 1-Wire® converter (DS2480B) that works as master.

Particle/Node. Consecutive particles share a dual-channel-addressable-switch (DS2413) encased in the rigid edge in between them, which acts as a slave on the 1-Wire® bus. To control the status of a particle, the MCU controls the output pins of its two switches, with direct access to their memory. As depicted in Fig. 3, each output pin drives a half-H bridge connected to a pair of SMA straps: neighbouring nodes, connected to the same pair of SMA straps, form a full-H bridge. By controlling the polarity of the H-Bridge (i.e., one half-H bridge pulls up and the other one pulls down) and introducing two diodes in mutual exclusion in series with the straps, each strap (SMA_L for left edge and SMA_R for right edge) can be activated independently.

Bus commutator. As the maximum acceptable current on the 1-Wire® bus is 5.4 mA, while the SMA straps nominally draw 180 mA, a commutator decouples data line and power supply (Vcc). To obtain the same decoupling also within each particle, a zener diode (5.6 V) in series with the power circuit prevents the straps from being supplied when the bus is in data mode. Instead, when the external commutator switches the line to Vcc (15–20 V), all the straps previously enabled are powered. The local capacitor on each DS2413 switch preserves the settings during the voltage transition. Also, the device can withstand 28 V.

C. Node Discovery and Localization

Each DS2413 is addressed by a factory lasered 64-bit ID. Since these addresses are unknown until the device is connected to the bus, a discovery algorithm is needed so that the MCU can find out all the addressable switches in the chain. The DS2480B bus master provides this functionality. However, the discovery algorithm does not indicate the actual order of the switches along the bus, i.e., the order of the particles. Since this is fundamental for controlling the straps and the entire chain, we devise a localization algorithm to identify which pair of switches can control a particle, in order to infer the relationships among particles.

The algorithm exploits the fact that when the two switches of a particle are active the MCU can detect an increase of power draw over the 1-wire bus, due to the current flowing through the SMA wire controlled by the two switches. In addition, we can assume that nodes are regularly deployed along the chain with the same orientation, as all half H-Bridges connected to pin 1 (2) of the switch connect to the particle above (below) except for the particles at the end of the chain (see Fig. 3).

The MCU hence picks an arbitrary node A_i (addressable switch), configures one half H-bridge (e.g., the one connected to pin 2 of DS2413) and then sequentially configures the complementary half H-bridges of all other nodes (e.g., the ones connected to pin 1) until power draw varies, in which case the two addressable switches belong to the same particle, therefore the node A_{i+1} is found. The two active addressable switches are reset, and the procedure is repeated for the newly-found neighbor node. If no neighbor can be found, a node is at the end of the chain, and all the nodes A_j such that $j \geqslant i$ have been found.

If no nodes remain to be localized, the node A_i becomes A_0 and all other indexes are coherently updated. Otherwise, the MCU configures the half H-Bridge of node A_i that was not previously configured and then sequentially configures the complementary half H-Bridges of all non-localized nodes until power draw varies, in order to identify nodes at the locations A_j such that $j < i$. In this way, the algorithm can also tell which particles are head and tail of the chain. For convenience, we enumerate each particle considering the lowest index of its two nodes (e.g., particle p_i contains nodes A_i and A_{i+1}, whereby A_i has the lowest index, as shown in Fig. 3).

D. Control

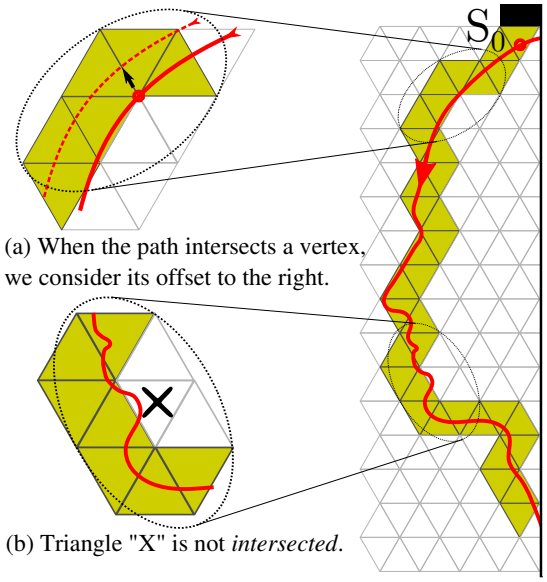
Folding the chain requires the MCU to synchronize a set of N folding particles p_i and the main actuator. The process consists of four steps characterized by time constants:

1) The 1-Wire master initializes the communication synchronizing the slaves while discovering their addresses taking time t_{setup};
2) The MCU sends the desired switch configuration to k addressable switches taking time t_{config} for each;
3) The MCU switches the bus to power supply mode to activate the SMA taking a minimum time t_{SMA} to fully contract the SMA;
4) The MCU activates the external actuator until the chain is folded taking time t_{mot}.

If we assume that the folding operation involves N particles, for which the configuration of k addressable switches needs to be set, the total time to perform the folding is:

$$T(N,k) = t_{setup} + k \cdot t_{config} + t_{SMA} + N \cdot t_{mot} \quad (1)$$

As a result of the regular geometry of the chain, the number of revolutions the motor performs to deform each particle is constant. Then also t_{mot} is constant for a constant speed.



(a) When the path intersects a vertex, we consider its offset to the right.

(b) Triangle "X" is not *intersected*.

Fig. 4: For each possible folding configuration of a chain, the particles lie on a regular triangular grid.

E. Planning

Planning consists in computing for each particle in a chain whether it should be folded left or right such that the folded chain approximates a given curve S that is to be displayed.

We observe that for each possible folding configuration of a chain, the folded triangular particles lie on a regular triangular grid as shown in Fig. 4, where the desired curve S resembles a head. We further assume that the chain hangs on a wall where the top particle is fixed to a wall bracket (black rectangle).

As multiple chains have to fold simultaneously we need to make sure that the same number of particles are folding among all chains. For this reasons, even in case the target shape is a vertical or a horizontal straight line, for which the unfolded configuration could be an option for a single chain, the goal is still to find a configuration where all the particles are folded.

We choose the top most point S_0 of S, and from there start to walk along S to create the sequence of triangles intersecting S. A triangle t_i is said to intersect S if S enters t_i through edge e_i and eventually leaves t_i through a different edge $l_i \neq e_i$. For example, in zoomed Fig. 4(b), the path intersects an edge of triangle “X” and then leaves the triangle through the same edge, therefore, the triangle marked “X” does not intersect. In case the path S intersects a vertex on the grid, generating an undefined situation, we consider those triangles that the path would intersect, if it had a slight offset to the right (w.r.t the walking direction), as shown in Fig. 4(a). Each intersecting triangle t_i corresponds to particle p_i in the chain. Likewise, each pair of edges e_i and l_i corresponds to the opposite rigid edges of particle p_i. Since we have chosen only adjacent intersecting triangles, the sequence of corresponding particles is also connected. To decide whether particle p_i should fold left or right, we can walk again along the path S and consider the side where the corner connecting e_i and l_i lies: if it lies on the left of S, the particle folds the left edge; if it lies on the right, the particles folds the right edge.

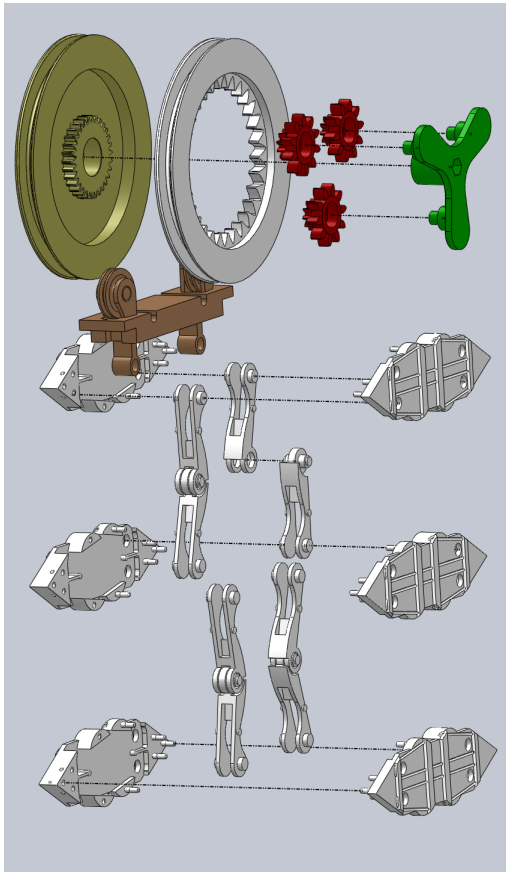


Fig. 5: An exploded view showing the main mechanical components of our prototype. The parts are 3D-printed and assembled together.

VI. Prototype

We have built a prototype of a force-guiding particle chain consisting of three particles. All the components of the chain shown in Fig. 5 are 3D-printed. The resistance and the flexibility of the used material PA22, along with the precision of the Selective-Laser-Syntering technology allowed us to realize the parts of a particle with snap-fit joints to simplify and speed up the assembly process.

Specifically, each rigid edge is composed of two symmetrical parts that can be snapped into each other and encase the printed circuit board with the electronics. Each half-link of a flexible edge is printed separately and then snapped into the complementary part. The flexible edge is then snapped into the two rigid edges. The SMA wire is threaded through a thin tunnel (diameter 0.5 mm) to extend the SMA path as shown in Fig. 2(c). We also investigated the possibility of printing pre-assembled hinges, aiming to improve joint stability and minimize assembly overhead, but the printing process requires a minimum distance between surfaces which causes unacceptable backlash.

The differential gear and spoolers are also 3D printed. One specific requirement for that component is that it should be thin enough to allow placing multiple chains next to each other. For that reason, we designed a planetary differential gear integrated with the spoolers. The planet gears and the carrier are printed separately, then assembled into the spoolers. Similarly, the gear connecting the motor and the encoder is a custom 3D-printed component.

VII. Evaluation

In this section we evaluate the design using the prototype described above. We first perform an experimental validation of the prototype and report on timing behavior and power consumption. We then analytically investigate the fundamental scaling properties of the design.

A. Experimental Validation

A PC connected via USB/UART to the MCU lets the operator control each particle individually and switches the bus between data and Vcc. We firstly checked the correct functioning of the 1-Wire® network, showing that the data line can be switched to Vcc for powering the SMA wires without issues for communication. We found that the 1-Wire master looses synchronization with the slaves after this operation, requiring a reset to restore the communication. This introduces a delay after each operation equivalent to t_{setup} (Sec. V-D). We measure that t_{setup} takes 205 ms with 6 nodes for the initialization of the bus; later resets can be performed within 78 ms, regardless of the number of nodes on the bus. The configuration of a node takes t_{config}=22 ms.

We installed a chain of three particles hanging from a custom support with the main actuator winding up the thread from the top. We empirically measured the activation time of the SMA to fully disengage the link (using video), and counted the actual revolutions of the spool to completely fold one particle. The SMA requires $t_{SMA} \geq 421$ ms to disengage the link when the system is supplied with 20 V and the SMA draws 280 mA (nominal value 180 mA). However, for safety reasons we increased this value to 500 ms.

Considering that to completely fold the chain the thread winds up by nominally 99.1 mm, the spooler (40 mm nominal diameter) requires 0.789 revolutions, which corresponds to approximately 113 cycles on the encoder (48 cycles/rev. connected to the shaft through a 1:3 gear). We reduce the motor speed using pulse width modulation. The time needed to actuate a single particle is then t_{mot}=1050 ms.

Inserting these values into Eq.1 and considering a single particle folding (i.e., two addressable switches have to be set), the time needed for this operation is $T(1,2) = 78\text{ms} + 2 \cdot 22\text{ms} + 500\text{ms} + 1 \cdot 1050\text{ms} = 1672\text{ms}$, which matches the experimental result we obtain. Actuating the system with the motor at the nominal speed of 14 RPM, t_{mot} reduces to 56 ms, for which $T(1,2)$ becomes 678 ms. The unfolding operation only requires 168 ms with three particles.

A complete actuation test, where all the particles are folded starting from the bottom and progressively proceeding to the top, demonstrated the correctness of the settings previously identified. Even though each deformation presents an error of ± 2 cycles on the encoder (mainly due to fabrication tolerance, but also due to the low accuracy of the measurements), the errors self-compensate when all the particles are folded.

The unfolding of the chain subject to gravity could not completely restore the initial configuration, although an additional rubber bands assist links to return to straight configuration. This happens because of the moderate weight of the components and especially due to the friction at the hinges. An additional weight of 16 g added to the tail of the chain improves the result, yet without completely unfolding the chain. Improvements in the manufacturing process to reduce the friction will address this issue.

B. Scalability

In this section we investigate the scaling limit of our approach, i.e., how many particles we can support in a single chain. For that we consider the worst-case situation depicted in Fig. 6, where N particles of a chain are already folded in a straight horizontal configuration (cantilever), while the $(N+1)$-th particle (top-right in the figure) is about to fold by applying F_s to the left flexible edge. We estimate the force F_w resulting on the pulling thread and compute the maximum number N of particles that can be supported such that the limited force F_s that can be exerted by the SMA is still sufficient to bend the flexible edge given F_w. Due to the leverage effect F_w is maximized in the shown configuration, therefore it constitutes the worst case. For this we first have to compute the force F_s that is required to bend a flexible edge when a pulling force F_w is applied to the particle:

$$F_s = 2 \, \tfrac{\delta}{h} \, F_w \tag{2}$$

where δ is the offset between the three hinges of the monostable link and h the distance between the two hinges at the extremes as shown in Fig. 2(b).

Inserting the actual dimensions of our prototype into Eq. 2, we can estimate the mechanical advantage F_w/F_s. With δ=1.28 mm and h=32.11 mm, the mechanical advantage is 12.5. A smaller δ would increase this value, but would also reduce the stability of the flexible edge.

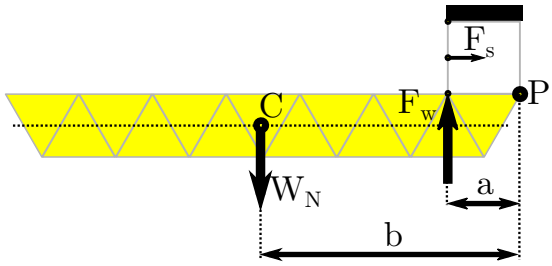


Fig. 6: Worst case configuration of a folded chain.

Now we can establish a relationship between the number N of particles and the resulting force F_w. For this we observe that the configuration shown in Fig. 6 can be considered a lever of class 3, where P is the fulcrum, F_w the effort, and W_N the resistance. We define W_N as the total weight of the cantilever applied to its centre of mass, $W_N = NW_p$, where W_p is the weight of one particle (about 4.5 g in our prototype). The two distances a and b correspond to the edge length l of a particle and $\frac{1}{4}Nl$, respectively. Applying the law of the lever we obtain:

$$F_w \cdot l = \tfrac{1}{4}Nl \cdot (NW_p) \tag{3}$$

We can now substitute F_w with $12.5 \cdot F_s$ according to the mechanical advantage computed above based on Eq. 2 and solve for N:

$$N = 2\sqrt{\frac{F_w}{W_p}} = 2\sqrt{\frac{12.5 \cdot F_s}{W_p}} \tag{4}$$

The maximum F_s an SMA strap can exert, depends on its contraction force: with a diameter of 100 μm the nominal contraction force F_{SMA} is 14.7 N. As the arrangement of the strap around the hinge of the flexible edge (Fig. 2) doubles the effect of its contraction force, while the angle the strap forms with the vector $\overline{F_s}$ reduces its effect by $cos(30°)$, we can approximate $F_s = 2F_{SMA} \cdot cos(30°)$ which corresponds to 2.59 kg. Substituting this value and $W_p = 4.5$ g into Eq. 4, we obtain $N = 84$ particles in the worst case. We observe that N is independent of the size l of the particles and grows with the inverse of the square root of the weight W_p for a constant F_s. This means that if we can miniaturize particles and reduce their weight, we can increase the number of particles in the chain in the worst case. Hence, the miniaturization is only limited by the mechanical stability of the thread and the material from which the particles are made.

In the best case (folded chain hanging down vertically) 8000 particles can be supported. Further work therefore includes the design of a planning algorithm that tries to avoid temporary folding states with a high F_w.

VIII. Conclusion And Future Work

We have presented a force-guiding particle chain that can fold into arbitrary flat curves under program control as a building block to construct a shape-shifting display that can fold its surface into a 3D shape that can be viewed, touched, and modified by users. The key feature of force-guiding particle chains is that the mechanical force to fold the chain is generated by an actuator external to the chain, such that the particles of the chain are simple and require only a mechanically weak actuator to guide the external force. Thereby, particles are amenable to miniaturization and scale up to a large number of particles per chain. We presented the mechanical and electrical design, as well as algorithms for control and planning. We demonstrate and validate a working prototype.

Future work includes, among others, improvement of the mechanical design to reduce friction and the number of parts; inclusion of sensors into particles for recognizing touch gestures and folding state of particles; as well as advanced planning algorithm to minimize F_w.

References

[1] S. C. Goldstein, J. Campbell, and T. C. Mowry, "Programmable matter," *IEEE Computer*, 2005.
[2] M. Weller, M. Gross, and S. Goldstein, "Hyperform specification: designing and interacting with self-reconfiguring materials," *Personal and Ubiquitous Computing*, 2011.
[3] H. Ishii, D. Lakatos, L. Bonanni, and J.-B. Labrune, "Radical atoms: beyond tangible bits, toward transformable materials," *Interactions*, 2012.
[4] M. Jorgensen, E. Ostergaard, and H. Lund, "Modular ATRON: modules for a self-reconfigurable robot," in *IROS*, 2004.
[5] B. Kirby, B. Aksak, S. C. Goldstein, J. F. Hoburg, T. C. Mowry, and P. Pillai, "A modular robotic system using magnetic force effectors," in *IROS*, 2007.
[6] S. Murata, H. Kurokawa, E. Yoshida, K. Tomita, and S. Kokaji, "A 3-d self-reconfigurable structure," in *Robotics and Automation*, 1998.
[7] V. Zykov, A. Chan, and H. Lipson, "Molecubes: An open-source modular robotics kit," *IROS*, 2007.
[8] K. Gilpin, K. Kotay, D. Rus, and I. Vasilescu, "Miche: Modular shape formation by self-disassembly," *Int. J. Rob. Res.*, 2008.
[9] J. Romanishin, K. Gilpin, and D. Rus, "M-blocks: Momentum-driven, magnetic modular robots," in *IROS*, 2013.
[10] B. Kirby, J. D. Campbell, B. Aksak, P. Pillai, J. F. Hoburg, T. C. Mowry, and S. C. Goldstein, "Catoms: Moving robots without moving parts," in *AAAI (Robot Exhibition)*, 2005.
[11] E. Hawkes, B. An, N. M. Benbernou, H. Tanaka, S. Kim, E. D. Demaine, D. Rus, and R. J. Wood, "Programmable matter by folding," *National Academy of Sciences*, 2010.
[12] B. An, N. Benbernou, E. D. Demaine, and D. Rus, "Planning to fold multiple objects from a single self-folding sheet." *Robotica*, 2011.
[13] C. D. Onal, R. J. Wood, and D. Rus, "Towards printable robotics: Origami-inspired planar fabrication of three-dimensional mechanisms." in *ICRA*, 2011.
[14] N. Correll, C. D. Onal, H. Liang, E. Schoenfeld, and D. Rus, "Soft autonomous materials - using active elasticity and embedded distributed computation," in *12th International Symposium on Experimental Robotics*, 2010.
[15] A. Knaian, K. C. Cheung, M. B. Lobovsky, A. J. Oines, P. Schmidt-Nielsen, and N. Gershenfeld, "The milli-motein: A self-folding chain of programmable matter with a one centimeter module pitch." in *IROS'12*.
[16] V. Zykov, E. Mytilinaios, M. Desnoyer, and H. Lipson, "Evolved and designed self-reproducing modular robotics," *IEEE Transactions on Robotics*, 2007.
[17] P. White, M. Posner, and M. Yim, "Strength analysis of miniature folded right angle tetrahedron chain programmable matter," in *ICRA*, 2010.

Paper B

DOI: 10.1145/2695664.2695932
Link: http://dl.acm.org/citation.cfm?doid=2695664.2695932
ISBN: 978-1-4503-3196-8

Force Model of a Robotic Particle Chain for 3D Displays

Matteo Lasagni and Kay Römer
Institute for Technical Informatics
Graz University of Technology, Austria
{lasagni, roemer}@tugraz.at

ABSTRACT

We aim to construct a 3D screen – an initially flat 2D surface that can fold into the third dimension to display arbitrary three dimensional surface shapes. Our approach is based on chains of robotic particles that can be individually actuated to fold into a desired curve. This paper contributes a computational model of the forces acting on the robotic particles. Experimental results show that the model accurately predicts reality. The model forms the core of force-minimizing planning algorithms that compute a folding sequence approximating a given target curve while minimizing the actuation forces. The model is also instrumental for simulation tools that allow to study forces while executing a given folding sequence.

Categories and Subject Descriptors

I.6 [**Simulation and Modeling**]: General; I.2.9 [**Robotics**]: Miscellaneous

General Terms

Theory, Experimentation

Keywords

Modeling, Simulation, Programmable Matter

1. INTRODUCTION

The ultimate goal of our work is to construct a 3D screen, i.e., an initially flat 2D surface that can fold into the third dimension to display arbitrary three-dimensional surface shapes. Fig. 1 illustrates this concept, where the screen displays the target shape (blue head). The display consists of a set of chains placed next to each other where each chain can fold to approximate a given curve (i.e., a slice of the target shape that is to be displayed). Each chain in turn consists of tiny robotic particles that can be individually actuacted to fold.

In our previous work [14] we have designed and implemented a working prototype of this approach. One of the key goals is to scale chains to a large number of tiny particles in order to maximize the resolution of the 3D display. This has been achieved by a clever mechanical design where the actuator that folds the chain is external to the particles and a simple locking mechanism in each particle controls how the external actuation force folds the particle. With this design, the number of particles in a chain is limited by the maximum force this external actuator can exert on the chain of particles.

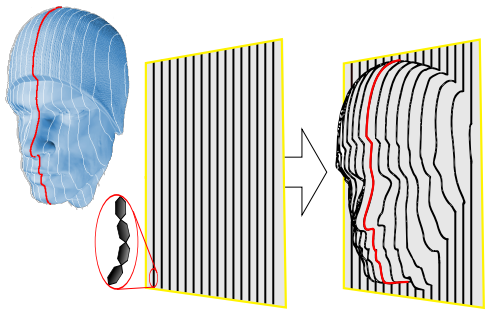


Figure 1: Basic principle: foldable chains made of tiny robotic particles form a screen that can take on the shape of a target 3D model (head).

However, the required actuation force (and thereby the maximum number of particles in a chain) is also a function of the sequence of folding operations. For example, lifting up a long chain of particles requires strong forces due to leverage effects. Unless the desired target shape requires such a folding configuration, intermediate folding states should be chosen such that actuation forces are minimized.

This leads to the concept of a *force-minimizing planning algorithm* which takes a desired target shape as input and computes a sequence of particle folding operations that result in the chain approximating the desired target while *minimizing actuation forces* during the execution of these folding operations.

Realizing such force-minimizing planning algorithms requires computational models of the forces acting on a particle chain. Firstly, such models are an integral part of the planning algorithm which essentially requires solving an optimization problem formulated on top of the model. Secondly, such models form also the core of simulation tools that allow to evaluate the forces occuring during the execu-

SAC'15 April 13-17, 2015, Salamanca, Spain.

http://dx.doi.org/10.1145/2695664.2695932

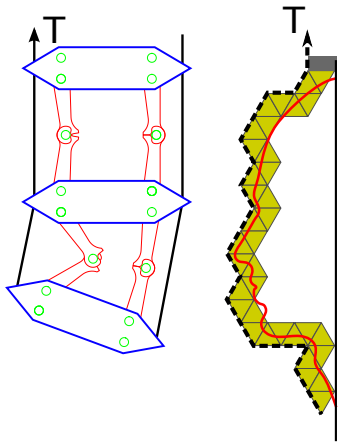


Figure 2: Working Principle

tion of a folding plan for different target shapes.

The contribution of this paper is the derivation of such a model that allows the computation of forces for a given folding state, but also the prediction of the progress of a folding operation when a given actuation force is applied.

We begin the paper with a brief summary of the hardware design of a particle chain from our previous work [14], before we introduce a force model of this design. We then report on experiments to confirm that the model accurately predicts reality. Related work is summarized before we conclude the paper.

2. BACKGROUND

In order to make the paper self-contained, we summarize in this section the design of a particle chain [14] that we set out to model in the next section. This design is driven by the requirement to scale chains to many tiny particles in order to achieve a high display resolution. This requires to minimize the mechanical complexity of particles so as to enable their miniaturization and manufacturing at low cost.

In [14], we therefore presented a chain of deformable particles that can outline arbitrary 2D shapes. To form a 3D display, multiple such chains are placed next to each other. Each of the chains folds to approximate a slice of the 3D object that is to be displayed. A chain consists of a sequence of particles with initially rectangular shape. A particle can be individually actuated to fold into triangles by compressing either its left or right vertical edge. Fig. 2 illustrates how such a folded chain of particles approximates the profile of a human head.

In order to simplify the mechanical design of particles and to enable their miniaturization, we took inspiration from underactuated robotic systems where passive elements are remotely actuated. In such systems, tensile forces propagate from the actuators by means of tendons throughout the robotic arm in order to remotely drive and actuate all the subcomponents. In this way, the subcomponents do not need local actuation but, instead, can be passive elements which in general present a very minimalistic design.

In a similar way, our system consists of robotic particles remotely actuated by a pair of counteracting tendons deployed along the sides of the chain and that exert a compressing force on each particle (as shown in Fig. 2 to the left). Particles present an initial quadrilateral shape that can turn into a triangle to provoke a local curvature of the chain. In particular, each particle can enable a deformation on one side by means of a simple built-in mechanism while, simultaneously, the external compressing force finalizes the folding action. This approach leads to a simple and minimalistic design of the particle, and ensures the scalability of the system, given that for a growing number of particles only the external actuator needs to be improved without affecting the size of the particles. Furthermore, by retaining the tendons, the chain maintains a desired posture, eliminating the need for latching mechanisms.

In our system, chains hang down from a wall bracket, as depicted in Fig. 2, and the tendons are tied to the tail of the chain and pulled up from the top-most particle.

3. FORCE MODELLING

A key goal of our work is to scale chains to large numbers of tiny particles in order to achieve a high display resolution. The scalability does not only depend on the mechanical design, but also on the *planning algorithm* which controls the sequence of folding actions, i.e., which particles are folded in which order. The reason for this is that during the folding process, leverage effects (e.g., lifting up a long chain of particles) may temporarily cause strong tensile forces even though the final folded configuration requires much smaller tensile forces. As there is a limit on the maximum tensile force supported by the chains, the planning algorithm has a strong influence on the scalability. This leads to the concept of a *force-minimizing planning algorithm* which computes a folding sequence that minimizes the tensile forces during the folding process.

In order to support the design of such force-minimizing planning algorithms, but also to simulate a given planning algorithm to study the tensile forces during the folding of a target shape, we require a model of the tensile forces that result in a given folding state of the chain.

Therefore, we aim to provide a physical model that tells us how forces redistribute across the system in equilibrium. Specifically, we estimate the actuation tension needed to deform the i-th particle for a specific system configuration assuming all the other particles in a frozen state and under the effect of gravity. For example, with respect to the chain shown in the right of Fig. 3, aiming to estimate the force acting on the highlighted particle i, we consider the tail of the chain as an undeformable block, including the particle $i + q$ although deforming at the same time. As particles have a mass that is not negligible, we need to model the mass distribution in order to estimate the moment of force due to gravity which counteracts the actuation force to deform particle i. To simplify the model, we represent the equation of the forces acting on a particle with respect to a local coordinate system centred at the middle point of its upper base.

When multiple particles are deforming simultaneously, the "frozen state" assumption is sufficient to characterize the forces acting on a given particle, but it is not possible to predict how the system will evolve. Indeed, if particles i and $i+q$ are both deforming at the same time, then all the particles in between, subject to gravity, have to be taken

into account since they are also free to move, despite not being deformed. We therefore improve our initial model by proposing a tension propagation model which addresses situations where multiple particles fold at the same time.

3.1 Assumptions

In the remainder of the section, we consider only the skeleton of a rectangular particle having the four edges hinged at the vertices. While this is an imperfect model of reality (as the non-negligible thickness of real hinges does not allow for two edges sharing a single hinge), this assumption significantly simplifies the model without introducing notable inaccuracies. We also assume that:

1. A particle initially presents a rectangular shape. Once fully deformed it outlines a triangle.
2. Particle deformation is regular in the sense that the two opposite angles, that each lateral edge forms with the horizontal edges, are identical; in other words, a folding particle always outlines an isosceles trapezoid, where the equivalent edges correspond to the top and the bottom base of the particle.
3. When a particle is folding into a triangle, we assume that the *compressing edge* is completely removed and does not interfere with the acting forces.
4. Friction at the hinges can be neglected, as we consider the quasi-static behaviour of the chain.
5. The two counteracting tendons present equivalent tension, as the applied forces are balanced.
6. Tendons pass through eyelets that are modelled as points.
7. When analysing the effect a tendon provokes on one side of a particle, even though the opposite tendon is simultaneously exerting a tension on the other side of the particle as well, we ignore the latter effect. This is equivalent to consider eyelets and hinges coincident.

3.2 Coordinate Systems

In order to simplify the model, we introduce a local coordinate system for each particle rather than considering a single global coordinate system for the whole chain of particles. For a given particle, this coordinate system is centered at the mid point of its upper base, with one axis pointing right and the other down. With this approach a particle deformation is equivalent to a change of basis between the coordinate systems of adjacent particles. In this way, we can also represent the tail of the chain starting from the particle i, with respect to the local coordinate system of i, ignoring all the particles above.

As depicted in Fig. 3, we formally define a coordinate system $\Omega_i \equiv (j_i, k_i)$ centred as the mid point of the upper base of the i-th particle with the j_i axis aligned to the base itself and the k_i-axis pointing down. Similarly, for the $(i+1)$-th particle which shares an edge with the i-th particle, we define a coordinate system $\Omega_{i+1} \equiv (j_{i+1}, k_{i+1})$. We can then derive a transformation matrix $M^{\Omega_i}_{\Omega_{i+1}}$ that is a function of the deformation angle α_i, and which transforms one basis into the other.

The matrix $M^{\Omega_i}_{\Omega_{i+1}}$ is a homogeneous transformation that combines a rotation $R^{\Omega_i}_{\Omega_{i+1}}$ and a translation $T^{\Omega_i}_{\Omega_{i+1}}$. Starting form the base Ω_i, indeed, the base Ω_{i+1} can be obtained by rotating the former by two times α_i (the sign depends on whether folding to the left or to the right) and by applying a shift of the vector $\overrightarrow{Si}$ (as shown in Fig. 3) with respect to Ω_i.

As by construction the vector $\overrightarrow{Si}$ encloses the angle α_i with the k_i-axis, we can define the vector on the basis of the versor $[sin(\alpha_i), cos(\alpha_i)]$ and the module $h - w \cdot |sin(\alpha_i)|$, where w is the length of a rigid edge of the particle (width of the chain), and h is the height of a particle.

$$\overrightarrow{Si} = (h - w \cdot |sin(\alpha_i)|) \cdot [sin(\alpha_i), cos(\alpha_i)] \tag{1}$$

The matrices governing the translation and the rotation from Ω_i to Ω_{i+1} are defined as

$$T^{\Omega_i}_{\Omega_{i+1}} = \begin{bmatrix} 1 & 0 & \overrightarrow{Si}_x \\ 0 & 1 & \overrightarrow{Si}_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$R^{\Omega_i}_{\Omega_{i+1}} = \begin{bmatrix} cos(2\alpha_i) & sin(2\alpha_i) & 0 \\ -sin(2\alpha_i) & cos(2\alpha_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

where $\overrightarrow{Si}_x$ and $\overrightarrow{Si}_y$ are the components of the vector $\overrightarrow{Si}$, and α_i is the deformation angle already introduced.
To combine all the above matrices together we define the matrix M that describes the deformation of a single particle and depends on the parameter α_i.

$$M^{\Omega_i}_{\Omega_{i+1}} = T^{\Omega_i}_{\Omega_{i+1}} \cdot R^{\Omega_i}_{\Omega_{i+1}} \tag{4}$$

3.3 Center of Mass

In order to model the tensile forces generated by gravity acting on a sequence of particles, we compute the center of mass of this sequence of particles. Without loss of generality, we concentrate the mass of a particle in one single point $\overline{C}$ whose position depends on the actual deformation of the particle. We then define a function $\overrightarrow{f_C(\alpha_i)}$ that indicates the location of the center of mass of the i-th particle with respect to its basis Ω_i when the deformation α_i is applied. Note that the function $\overrightarrow{f_C(\alpha_i)}$ could be generally approximated with the geometrical barycentre of the particle. However, to better represent asymmetric mechanical designs, for which the center of mass may not necessarily coincide with the geometrical barycentre of the particle, the function has to be defined in a design-specific manner.

Once we know the α_i values and the function f_C, we can estimate the center of mass of the entire chain or of a group of adjacent particles. In order to calculate the forces acting on the upper edge of the i-th particle due to weight of the chain tail, we can compute the center of mass of the tail starting from the i-th particle with respect to the basis Ω_i and multiply this value by the total mass of the particles involved and the gravitational acceleration.

If we denote R_i as the center of mass of the sub-chain starting from the particle i extending to the last particle N, with respect to the coordinate system Ω_i, we can compute R_i as follows:

$$\overrightarrow{R_i} = \frac{1}{N - i + 1} \sum_{t=i}^{N} M^{\Omega_i}_{\Omega_t} \overrightarrow{f_C(\alpha_t)} \tag{5}$$

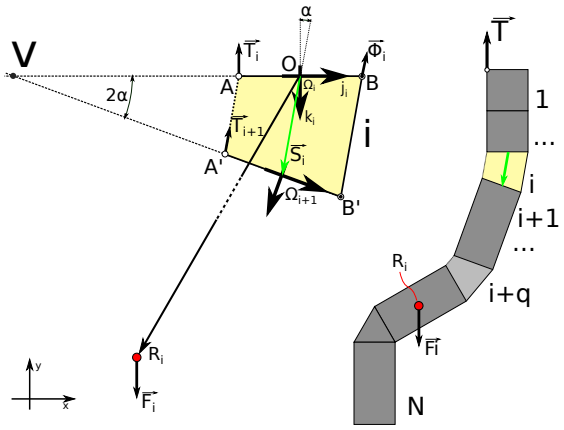


Figure 3: Forces acting on the chain at the equilibrium.

where $M_{\Omega_t}^{\Omega_i}$ concatenates all the matrices to transform the basis Ω_i into the reference basis Ω_t, defined as follows:

$$M_{\Omega_t}^{\Omega_i} = \begin{cases} M_{\Omega_{i+1}}^{\Omega_i} \cdots M_{\Omega_t}^{\Omega_{t-1}} & \text{if } t > i \\ diag(1,1,1) & \text{if } t = i \end{cases} \quad (6)$$

3.4 Actuation Tension

Now we can compute the actuation force T required to hold a chain in a given folding state in static equilibrium. For this we assume a folding edge of particle i as if it did not exist such that it does not interfere with the action of the tendon. Therefore, the only force counterbalancing the actuation force T_i (see Fig. 3) is the weight of the tail of the chain from the i-th particle to the end of the chain, which equals the force F_i due to gravity applied to the center of mass R_i, previously calculated with respect to the basis Ω_i.

As all particles have the same mass m, the force F_i can be computed as:

$$\overrightarrow{F_i} = (N - i + 1) \cdot m \cdot \overrightarrow{g_i} \quad (7)$$

where m is the mass of a particle and $\overrightarrow{g_i}$ the gravitational acceleration w.r.t. the base Ω_i (we compute $\overrightarrow{g_i}$ by rotating the vector $\overrightarrow{g}$ defined w.r.t. the absolute coordinate system (x, y) by an angle $\pi - 2\sum_{t=0}^{i-1} \alpha_t$).

We can now calculate the tension force needed to actuate the system to deform the particle i by solving the following equation system (which describes the system at the equilibrium) for T_i:

$$\overline{R_i} \times \overrightarrow{F_i} + \overline{OA} \times \overrightarrow{T_i} + \overline{OB} \times \overrightarrow{\Phi_i} = \overrightarrow{0} \quad (8)$$

$$\overrightarrow{F_i} + \overrightarrow{T_i} + \overrightarrow{\Phi_i} = \overrightarrow{0} \quad (9)$$

where $\overline{OA}$ and $\overline{OB}$ are the segments forming the upper base of particle i, as shown in Fig. 3, and Φ_i is the force applied at the hinge B. As in Eq. 8 Φ_i can be replaced with $-\overrightarrow{F_i} - \overrightarrow{T_i}$ from Eq. 9, we can solve the system for $\overrightarrow{T_i}$.

3.5 Tension Propagation Model

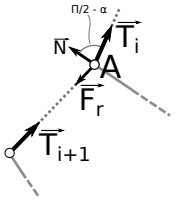


Figure 4: Tension propagation and friction

To compute the tension T_{i+1} acting at the $(i+1)$-th particle, we assume that the friction force $\overrightarrow{F_r}$ (see Fig. 4) at any eyelet is proportional to the normal force $\overrightarrow{N}$ (parallel to the bottom base of particle i) by a factor μ that depends on the material used for the tendon and the particle. We also assume that each eyelet is a single point and all the forces are concentrated on that single point. As the friction force $\overrightarrow{F_r}$ works against the tensile force of the tendon, the propagation model can be represented as:

$$T_{i+1} = T_i - F_r = T_i(1 - \mu \cdot |sin(\alpha_i)|) \quad (10)$$

in which we have expressed only the modules of the acting forces.

If more than one particle is folding at the same time, by means of the above model we can compute how forces redistribute among particles and which particle tends to complete its deformation first. For example, when particles i and $i + q$ are deforming simultaneously, if the sum of the friction forces at the eyelets along the tendon path is larger than the tensile force T_{i+1} (needed to actuate the sub-chain $[i + 1..N]$) then particle i deforms first; otherwise, particle $i+q$ deforms until either the previous condition holds or the deformation is completed.

4. MODEL VALIDATION

In order to assess the accuracy of the presented model, we measure the force T that a tendon exerts to maintain a chain in static equilibrium. We extend a chain of two particles with a piece of plastic material in order to simulate the presence of a longer tail. As the piece of plastic has rectangular shape, the calculation of the center of mass is straight forward.

The top base of the first particle is fixed to a horizontal bracket (see Fig. 5). An auxiliary arm at a distance K above the bracket holds a spring to which we connect the tendon under test. By means of a perforated strip we can shorten the distance S between the spring and the arm in order to modulate the force applied to the tendon, which is proportional to the elongation ΔL of the spring (unloaded length L_0).

For this experiment, we enable only the deformation of the second particle, while the head particle keeps the default rectangular shape. The experiment proceeds by progressively increasing the force applied to the tendon (i.e., by iteratively reducing S by multiples of a constant ΔS that is the distance between the strip holes) and measuring the distance d between the head of the tendon and the first particle. As the sum of the three lengths $S - \Delta S + L_0 + \Delta L + d = K$ is constant, we can calculate the value ΔL, which is proportional to the tension T, as a function of ΔS and d. We prefer

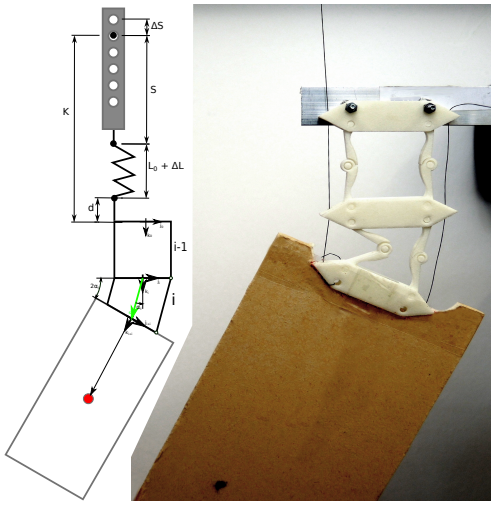


Figure 5: Experimental setup

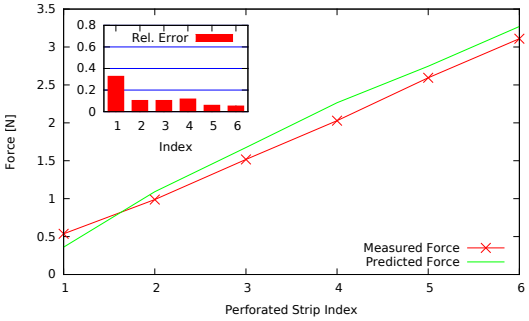


Figure 6: Evaluation of the model.

to measure d instead of directly measuring ΔL, because for the specific setup d can be measured more precisely.

Although in our model the values of $\overline{OA}$ and $\overline{OB}$ are identical, the actual particle prototype presents an offset between the hinge and the eyelet. Nevertheless, since in Eq. 8 the two parameters are kept separate, we can define each of them independently to better comply with reality. Specifically, $|\overline{OA}| = 19.9\,mm$ and $|\overline{OB}| = 12.83\,mm$. The dimensions of the undeformed particle are measured considering the rectangle that the four hinges outline, for which $w = 2|\overline{OB}|$ and $h = 31.43\,mm$.

As earlier explained, to reproduce the effect of a longer chain, we attach an extra mass to the tail of the chain. To simplify calculation, we express the center of this extra mass with respect to the coordinate system Ω_{i+1} (see Fig. 3) assuming i is the last particle of our chain and the extra mass represents particles $[i+1..N]$. The vertical axes of symmetry of both the rectangle and the particle are conveniently aligned. It follows that $R_{i+1} \equiv (0, 292.5)\ [mm]$, given the dimension of the rectangle $8.3\,cm \times 58.5\,cm$ and its mass $60\,g$ (which includes also the bottom base of the particle).

4.1 Results

With respect to the diagram shown in Fig. 5, for each possible ΔS step ($\Delta S = 9.05\,mm$), we measure the corresponding distance d between the top base of the head particle and the bottom end of the spring, and we derive ΔL as we know parameter values K and L_0. We have verified that the spring is linear with an elastic coefficient $k = 9.94\,g/mm$ which allows as to compute the tensile force T for a given elongation ΔL. We can then compare the actual force with the expected force T_i predicted by our model.

To compute the expected force T_i, we need to know the deformation angle of the particle under test. With reference to Fig. 3, we can calculate the distance $\overline{AA'}$ as the difference between d and d_0 initially measured when the particle was not deformed. In addition, as we know the structural dimensions of the actual particle, and specifically $\overline{AB}$ and $\overline{BB'} = 31.43\,mm$, we can compute the unknown angle by solving the following system of equations for α (see Fig. 3):

$$\begin{cases} \overline{AA'} = \overline{VA} \cdot sin(\alpha) = d - d_0 \\ \overline{BB'} = \overline{VB} \cdot sin(\alpha) \end{cases} \tag{11}$$

We calculate $\overline{R_i} \times \overrightarrow{F_i}$ since we know R_{i+1} and the extra mass, and solve Eq. 8 and Eq. 9 for T_i.

The two graphs in Fig. 6 show for each step ΔS a comparison between the measured and the predicted force as well as the relative error. We notice that, apart from an initial divergence, the model predicts the behaviour of the real prototype with a relative error below 0.12.

We conjecture that the initial divergence is due to the backlash at the mechanical joints of the prototype, which introduces a non-linearity into the system. Indeed, if the applied force is not sufficient to completely lift up the lower particle, which is in fact the case for smaller forces applied during the experiment, in the calculation of the deformation angle α the value d is affected by an offset due to the backlash among parts. We also consider that being the measured force small, the relative error tends to grow significantly, despite the absolute error remains in the average.

5. RELATED WORK

Earlier research on programmable matter has shown how ensembles of modular robotic particles can embody the underlying infrastructure to support the creation of shape-shifting materials. Indeed, by altering the reciprocal connection among particles and hence dynamically re-arranging their configuration, it is possible to control the overall shape that particles form together [6, 19, 9].

We can generally identify two main design principles that differ on the modality particles interact with each other and from which derives the freedom that particles have to rearrange. In one case, for example, modular robots are so highly dynamic and autonomous to the extent that they can temporary detach from neighboring particles and "climb" each other to reach new locations within the ensemble [10, 12, 16, 21, 5, 18]. The second case, instead, envisions robotic particles inherently built in the structure of the system which are able to provoke a deformation of the structure itself in order to control the overall shape. Depending on the particles topology, which can be flat or linear, we can have substrates that fold along crease patterns like in Origami [8, 1, 17] or

chains that fold in 2D [3, 7] or even 3D [20, 13, 22].

Although, on the one hand, the first approach allows the formation of literally *any* shape, the complexity the modular robots present (i.e., due to sophisticated detachable mechanisms and actuators) can obstacle the particles miniaturization that is fundamental for our purpose, aiming to create a "high resolution" screen. On the other hand, in a foldable structure where robotic particles are bound to each other in a fixed topology, beside the inherent physical constraints (which, nevertheless, do not prevent from the formation of arbitrary shapes [7]), particles present a very minimalistic design that facilitates later miniaturization. However, the principal issue for both the approaches concerns the scalability of the systems in terms of number and size of elements, as growing actuation forces to support a larger system calls for stronger, hence bigger, actuators. Being the actuators built in the particles, also the size of the latter grows accordingly.

To overcome the problem, we took inspiration from underactuated tendon-driven robotics and in our previous work [14], we implemented a foldable chain where the main actuator is dislocated outside the particles and each particle driven by tendons. Several systems have been shown [15, 4, 11] where articulated infrastructure of passive chained elements are remotely actuated by means of tendons. For example, tiny manipulators in use for remote surgery exploit tendons to drive the robotic arms with the aim of reducing size and inertia of the arm and the end-effector [2].
Being the sole positioning of the end-effector not sufficient for our purpose, also the physical models presented in the referenced works are hardly adaptable to predict forces acting on our system, for which a custom model is necessary.

6. CONCLUSIONS

We have derived the computational model of the forces acting on a chain of robotic particles that can fold into a desired curve and which is the building block of a 3D screen that can display arbitrary shapes. Such a computational model becomes the core of new simulation tools that allow to evaluate the force acting on the folding chains while approximating a target shape. Consequently, novel *planning algorithms* can derive, which aim to minimize the actuation forces through the computation of optimal actuation strategies and for which the computation model is integral part.

To assess the accuracy of the presented model, we have compared the force a tendon exerts on a chain of particles, where an extra mass was artificially added to simulate the presence of a longer tail, against the force our model predicts for the chain under test. The results have shown that the model can accurately predict reality.

7. REFERENCES

[1] B. An, N. Benbernou, E. D. Demaine, and D. Rus. Planning to fold multiple objects from a single self-folding sheet. *Robotica*, 2011.

[2] F. Anooshahpour, I. G. Polushin, and R. V. Patel. Quasi-static modeling of the da vinci instrument. In *Proc. Intelligent Robots and Systems, IROS*, 2014.

[3] N. Correll, C. D. Onal, H. Liang, E. Schoenfeld, and D. Rus. Soft autonomous materials - using active elasticity and embedded distributed computation. In *12th Int. Symposium on Experimental Robotics*, 2010.

[4] V. Falkenhahn, T. Mahl, A. Hildebrandt, R. Neumann, and O. Sawodny. Dynamic modeling of constant curvature continuum robots using the euler-lagrange formalism. In *Proc. IROS*, 2014.

[5] K. Gilpin, K. Kotay, D. Rus, and I. Vasilescu. Miche: Modular shape formation by self-disassembly. *Int. Journal of Robotics Research*, 2008.

[6] S. C. Goldstein, J. Campbell, and T. C. Mowry. Programmable matter. *IEEE Computer*, 2005.

[7] S. Griffith, D. Goldwater, and J. M. Jacobson. Robotics: Self-replication from random parts. *Nature*, 2005.

[8] E. Hawkes, B. An, N. M. Benbernou, H. Tanaka, S. Kim, E. D. Demaine, D. Rus, and R. J. Wood. Programmable matter by folding. *National Academy of Sciences*, 2010.

[9] H. Ishii, D. Lakatos, L. Bonanni, and J.-B. Labrune. Radical atoms: beyond tangible bits, toward transformable materials. *Interactions*, 2012.

[10] M. Jorgensen, E. Ostergaard, and H. Lund. Modular ATRON: modules for a self-reconfigurable robot. In *Proc. Intelligent Robots and Systems, IROS*, 2004.

[11] T. Kato, I. Okumura, H. Kose, K. Takagi, and N. Hata. Extended kinematic mapping of tendon-driven continuum robot for neuroendoscopy. In *Proc. Intelligent Robots and Systems, IROS*, 2014.

[12] B. Kirby, B. Aksak, S. C. Goldstein, J. F. Hoburg, T. C. Mowry, and P. Pillai. A modular robotic system using magnetic force effectors. In *Proc. IROS*, 2007.

[13] A. Knaian, K. C. Cheung, M. B. Lobovsky, A. J. Oines, P. Schmidt-Nielsen, and N. Gershenfeld. The milli-motein: A self-folding chain of programmable matter with a one centimeter module pitch. In *Proc. Intelligent Robots and Systems, IROS*, 2012.

[14] M. Lasagni and K. Römer. Force-guiding particle chains for shape-shifting displays. In *Proc. Intelligent Robots and Systems, IROS*, 2014.

[15] M. J. Martell and J. A. Schultz. Multiport modeling of force and displacement in elastic transmissions for underactuated hands. In *Proc. IROS*, 2014.

[16] S. Murata, H. Kurokawa, E. Yoshida, K. Tomita, and S. Kokaji. A 3-d self-reconfigurable structure. In *Robotics and Automation*, 1998.

[17] C. D. Onal, R. J. Wood, and D. Rus. Towards printable robotics: Origami-inspired planar fabrication of three-dimensional mechanisms. In *ICRA*, 2011.

[18] J. Romanishin, K. Gilpin, and D. Rus. M-blocks: Momentum-driven, magnetic modular robots. In *Proc. Intelligent Robots and Systems, IROS*, 2013.

[19] M. Weller, M. Gross, and S. Goldstein. Hyperform specification: designing and interacting with self-reconfiguring materials. *Personal and Ubiquitous Computing*, 2011.

[20] P. White, M. Posner, and M. Yim. Strength analysis of miniature folded right angle tetrahedron chain programmable matter. In *Proc. ICRA*, 2010.

[21] V. Zykov, A. Chan, and H. Lipson. Molecubes: An open-source modular robotics kit. In *IROS*, 2007.

[22] V. Zykov, E. Mytilinaios, M. Desnoyer, and H. Lipson. Evolved and designed self-reproducing modular robotics. *IEEE Transactions on Robotics*, 2007.

Paper C

DOI: 10.1109/AIM.2016.7576993
Link: http://ieeexplore.ieee.org/document/7576993/
ISBN: 978-1-5090-2065-2

Programmable Robotic Chains: Kinematics and Dynamics of a Scalable Tendon-Driven Under-Actuated Multibody System

Matteo Lasagni and Kay Römer

***Abstract*— In previous work, we presented a programmable *shape shifting-surface* composed of modular robotic chains. As the control of a robotic chain presents many challenges, in this paper, kinematics and dynamics of such a robotic chain are modelled to enable *model-predictive* planning and control strategies. A robotic chain is a tendon-driven under-actuated multibody system that can piecewise control its curvature to approximate complicated 2D curves. As the actuation forces depend on the sequence of intermediate configurations to progressively achieve a target geometry – considering that too intense forces might compromise the stability and the integrity of the system – optimal planning strategies are expected to limit the maximum actuation forces. To this end, a model-predictive control process is required to properly actuate the system, for which a feedback-loop control is not possible due to the absence of sensors for low-cost and design reasons.**

Our contribution consists in the derivation of a dynamic model of the robotic chain to support model-predictive planning and control. An evaluation of the derived model proves its accuracy in comparison to an existing prototype.

I. Introduction

In [1], we presented the design of a programmable *shape-shifting surface*, a robotic device that can dynamically take on the shape of arbitrary 3D objects. Among the potential applications, we envision the creation of tangible 3D displays to help users better understand complicated 3D models. For example, a shape-shifting surface can help the designer to easily present the working-principle of his newly conceived device, dynamically zoom in on details, and hence facilitate the communication among heterogeneous domain-experts (e.g., engineers and investors).

Aiming at high resolution rendering, our solution consists of modular foldable robotic chains, whose curvature can be piecewise controlled to approximate a section of the target 3D object to display. The piecewise structure of a chain results from the concatenation of modular "robotic particles" that can be individually actuated to locally modify the curvature of the chain. As the resolution depends on size and number of particles, we devised a very minimalistic mechanical design to facilitate particle miniaturization and also to guarantee system scalability. Our solution results in an *under-actuated tendon-driven multibody system*.

A chain approximating a section of a head is depicted in Fig. 1a. The lower part of Fig. 1a shows the mechanical design of the chain, more precisely, of three concatenated

Matteo Lasagni and Kay Römer are with the Institute for Technical Informatics, Graz University of Technology, 8010 Graz, Austria {lasagni,roemer}@tugraz.at

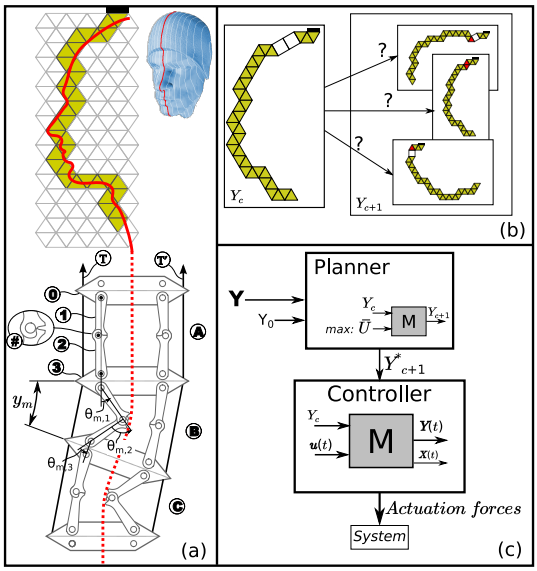


Fig. 1: System Principle and Model-predictive control.

particles. Particles in their initial state (e.g., particle **A**) present a squared geometry □ that can fold into an equilateral triangle either to the left ◁ (e.g., particle **B**) or to the right ▷ (e.g., particle **C**). This implies that chains are constrained to fold in the 2D space. As a folded particle (◁, ▷) is mechanically more stable than an unfolded one (□), in its final configuration a chain only admits fully folded particles. Consequently, considering that all the possible configurations of a chain lie on a regular triangular grid, as shown in Fig. 1a, it is straightforward to infer the final configuration of particles, given the target shape to approximate (i.e., a section of the target object). By overlapping the contour of the target shape on such a grid, the sequence of triangles the contour intersects, indicates how particles need to fold for the chain to approximate the contour [1]. This method intuitively shows that, despite only folded particles are admissible, the chain can approximate arbitrary shapes as long as their contour does not intersect the same triangle more than once, as detailed in [1].

A modular particle (e.g., **A** in Fig. 1a) consists of six links and six revolute joints. The lateral links (e.g., **1** and **2**) are designed to "lock" the particle in the initial state □. A slight misalignment among the three joints of links **1** and **2**, combined with the detent **#**, impedes the relative rotation of

the two links, thus the initial □ layout persists.
The transition between states □ → ◁ and □ → ▷ requires a slight inwards movement of joint **#** in order to "unlock" one of the two folding sides. However, as this action alone is not sufficient to fully fold a particle, a pair of externally actuated tendons (**T** and **T'**) exerts a compressing force on each particle in the chain. Such a force causes the lower link of an unlocked particle (e.g., link **3**) to move towards its upper link (e.g., link **0**) and thus fully folds into a triangle. In this way, an extension of the chain (i.e., an increment of the number of particles) only requires an upgrade of the external actuator, hence the system is scalable. Also, the tension applied to the tendons can be regulated to retain the final posture of the chain, thus not needing any latching mechanism in the particle. The absence of embedded bulky actuators and latches makes particles amenable for miniaturization.

While the tendon on the folding side of a particle shortens (i.e., is pulled up), the other tendon slightly extends. A differential-gear (installed between the external motor and the two spools winding up the tendons) balances the forces applied to the tendons and compensates for their opposing behavior. Furthermore, a differential gear installed on each chain, decouples the actuation among multiple chains, thus enables the use of a single actuator for the whole system [1]. This, however, constrains all the chains to fold exactly the same number of particles. It is worth noting that a particle becomes *under-actuated* as soon as one side is unlocked. In this condition, indeed, the three degrees of freedom of a particle depend on one single control input, that is the compressing action due to the tendons.

While having tiny particles is desirable, the combination of many of them makes the actuation of a chain challenging. A particle requires about $400\,mA$ for unlocking, hence it is not feasible to actuate all the particles simultaneously [2]. A possible solution to overcome this problem is to split the actuation process in two stages. In the first stage "optimal planning", a sequence of *intermediate configuration* is computed to progressively reach the final target configuration, where all particles are folded. At each intermediate configuration, a subgroup of particles is selected to fold among those still unfolded (□). One critical aspect is that inappropriate selections provoke undesirable peaks of forces, which can induce mechanical stress and compromise the stability and the integrity of the entire system. Therefore, the *planner* (Fig. 1c) searches among the possible *next* configurations (Fig. 1b, starting from $\boldsymbol{Y}_c$ and aiming at $\boldsymbol{Y}_{c+1}$) the one that limits actuation forces within a given bound. To this end, a model of the chain is required to predict its behavior upon the application of maximum admissible actuation forces $\bar{\boldsymbol{U}}$.

The second stage "optimal control" computes a set of *minimal forces* necessary to operate the transition between consecutive intermediate configurations $\boldsymbol{Y}_c \rightarrow \boldsymbol{Y}_{c+1}$. The absence of sensors in particles (for miniaturization and low-cost reasons), prevents any feedback from the system. Thus a *model-predictive controller* (Fig. 1c) computes such forces and ensures the controllability of the chain (i.e., prevent tendons from loosening). A model of the chain is required to predict its *dynamical* behavior upon the application of control forces $\boldsymbol{u}(t)$, starting from an initial condition $\boldsymbol{Y}_c$. Also, to ensure controllability, the model predicts the internal state of the system $\boldsymbol{X}(t)$, including the tension of the tendons.

Our contribution in this paper consists in the derivation of a dynamic model to predict the behaviour of the chain. As the latter is an under-actuated tendon-driven multibody system, the formulation of the dynamic model presents new challenges. Furthermore, aiming to later implement the model-predictive controller in the embedded hardware of a complete shape-shifting surface, a particular focus is on an efficient yet accurate model.

A. Background and Motivation

Existing physics simulation engines, able to model and predict the behaviour of *multibody systems*, are undoubtedly very powerful and general-purpose tools. Developed in a versatile manner to cope with multiple aspects including numerical stability [3], [4] and contained drift-off problem [5], they exhibit high performance by exploiting parallel computing on multi-core architectures. However, while the complexity of their software architecture hampers porting to embedded platforms, their computational requirements decisively prohibit their use in constrained embedded computing platforms. As remarked in [6], common simulation software and libraries mainly target applications in traditional industrial and robotic automation, with limited support for tendon-driven robotics. Consequently, aiming to build a model-predictive controller for the shape-shifting surface, this paper presents a model formulation explicitly suitable for the implementation on an embedded computing platform. With some opportune considerations and assumptions we can reduce the complexity while still ensuring acceptable accuracy. For example, as the prediction horizon is limited to the period between consecutive configurations $\boldsymbol{Y}_c \rightarrow \boldsymbol{Y}_{c+1}$, constraint stabilization [7] is not a critical aspect.

Efficient algorithms [8], [9], originally developed for open-chain and tree-type systems, have shown to be applicable also to closed-loop systems. Recursive methods [8]–[11] based on Newton-Euler formulation, systematically iterate over the nested topology of a multibody system to progressively compute the dynamics of each component. They have been proven to be quite efficient with complexity $O(n)$ when applied to open-chain [12], [13] and tree-type systems [11]. The same algorithms combined with Langrage multipliers [4] apply to closed-loop systems as well.

The general assumption of recursive methods is that linear forces and torques (wrenches) act locally at the joints [14], [15], and also that the dynamics of a body are influenced by the adjoining bodies – e.g., Shah et al. [11] formulated a wrench propagation matrix to characterize the reciprocal influences of consecutive bodies. In tendon-driven robotics, where forces act in a more distributed fashion, the mutual interactions among non-adjoining bodies, due to tendons, need to be considered. For example, the two tendons (Fig. 1a) induce an interaction between links 0 and 3, even though the latter are separated by links 1 and 2. To model this

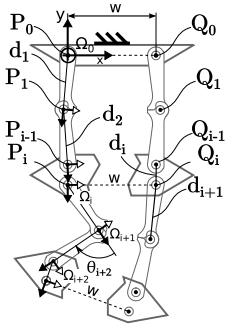


Fig. 2: Unconstrained topology.

situation, we consider the line of action of the tensile forces and then compute their effect on the bodies involved. The tension propagation model presented in our earlier work [2] estimates the friction between tendons and eyelets and the subsequent loss of tension.

Lee et al. proposed a recursive method [16] specific for tendon driven-robots. However, as they assume pulleys at each rotary link as well as pulleys at each *intermediate* link to guide the tendons, their system presents only forces acting locally at the joints. Furthermore, as an unlocked particle results in an under-actuated tendon-driven system, a novel model approach is required.

II. Model Overview

This section introduces the requirements to drive the subsequent kinematic and dynamic models formulation. As the final model predicts the behavior of a chain subject to actuation forces, input and output of the model are defined. Since the differential gear part of a chain decouples its actuation from the other chains in the system, in the remainder we only focus on a single chain assuming the model of the whole system can result from the combination of the models of its chains.

A. Model Requirement

The two models "M" shown in Fig. 1c can be formulated in an equivalent manner. In the *planner* a model predicts whether a transition between intermediate configurations $\boldsymbol{Y}_c \rightarrow \boldsymbol{Y}_{c+1}$ is *feasible* upon the application of maximum actuation forces $\bar{\boldsymbol{U}}$. Although a quasi-static model [2] could suffice to predict the behavior of a short chain, in a long chain the inertial forces need to be considered, for example, if the actuation induces a swinging effect on the tail.

Similarly, in the *controller* (Fig. 1) the model predicts the behaviour of a chain during the transition $\boldsymbol{Y}_c = \boldsymbol{Y}_{c+1}$. Here the prediction is done over time $\boldsymbol{Y}(t)$ and also the internal state $\boldsymbol{X}(t)$ is computed. Goal of the controller is to first seek the minimal actuation forces $\boldsymbol{u}(t)$ to eventually make the chain reach the next intermediate configuration $\boldsymbol{Y}(t^*) \equiv \boldsymbol{Y}_{c+1}$ and then actuate the plant with such $\boldsymbol{u}(t)$.

In this manner we can formulate a single dynamic model of the chain and apply it to both cases. In the following, we assume the sequence of intermediate configurations to be known. We focus on the generic transition $\boldsymbol{Y}_c \rightarrow \boldsymbol{Y}_{c+1}$ and derive the model to predict $\boldsymbol{Y}(t)$ over time.

B. Chain Actuation

In order to discover the new intermediate configuration $\boldsymbol{Y}_c \rightarrow \boldsymbol{Y}_{c+1}$, the vector $\bar{\boldsymbol{U}}$ indicates the maximum admissible actuation forces during the transition $\boldsymbol{Y}_c \rightarrow \boldsymbol{Y}_{c+1}$. The *planner* discovers which next intermediate configuration is feasible under this condition. Assuming the next intermediate configuration $\boldsymbol{Y}_{c+1}$ is known, the vector $\boldsymbol{u}(t)$ indicates the intensity of the actuation forces at each time t to *control* the chain while performing the transition $\boldsymbol{Y}_c \rightarrow \boldsymbol{Y}_{c+1}$. As $\boldsymbol{u}(t)$ is a generalization of $\bar{\boldsymbol{U}}$ and the same model is used for both planning and control, we focus on the second case.

During the transition $\boldsymbol{Y}_c \rightarrow \boldsymbol{Y}_{c+1}$ a subset of unfolded particles folds. The vector $\boldsymbol{Y}_c = [\cdots y_m \cdots]$ indicates for every particle s the current state $y_m \in \{\square, \triangleright, \triangleleft\}$. Referring to Fig. 3, a combination of forces is required to first unlock and then fold a particle. The forces f_m and f'_m select the folding side, while the tensile forces T and T' compress the particle until the latter is completely folded. This operation modifies the curvature of the chain $\boldsymbol{Y}_c \rightarrow \boldsymbol{Y}_{c+1}$. The vector $\boldsymbol{u}(t)$ indicates at each time t and for each particle s the intensity of the forces f_m, f'_m, T and T'.

C. Internal State Representation

As a particle has more than one degree of freedom (DOF), the chain configuration vector $\boldsymbol{Y}(t)$, where each element $y_m(t)$ indicates the angle enclosed by the upper and the lower link of a particle, namely the state of particle m over time, is not appropriate to formulate the subsequent kinematic model as it does not fully characterize the chain. A vector $\boldsymbol{\theta}(t)$ is defined to represent the "internal state" of the chain, in terms of relative-joint coordinates – i.e., angles enclosed by adjoining links.

Regardless of the angular limits due to the specific design, the loop-closure of a particle allows three DOF. This means, three independent coordinates fully define its state. Referring to Fig. 1, we conveniently choose three joint-coordinates $\theta_{m,1}(t)$, $\theta_{m,2}(t)$ and $\theta_{m,3}(t)$ that fully define particle **B** at each time t. Their linear combination corresponds to the angle $y_m(t)$. In an M-particle chain, particles are enumerated starting from $m = 1$ (the topmost particle) and proceeding top-down until particle M. The vector $\boldsymbol{\theta} = [\cdots, \theta_{m,1}, \theta_{m,2}, \theta_{m,3}, \cdots,]$ defines the internal state of the whole chain and directly maps to the chain configuration vector: $\boldsymbol{\theta}(t) \mapsto \boldsymbol{Y}(t)$. While $\boldsymbol{\theta}(t) \mapsto \boldsymbol{Y}(t)$ is defined for any t, the relationship $\boldsymbol{Y}(t) \mapsto \boldsymbol{\theta}(t)$ is only defined for the *initial*, *intermediate*, and *final* configurations where particles are in one of the "discrete" states $\{\triangleright, \triangleleft, \square\}$.

III. Kinematic Model

This section extends the concept of internal state in order to describe the complete geometry of the chain and its constraints. As the loop-closure of particles implies implicit kinematic equations, we initially ignore the loop-closure

and identify two independent branches P and Q whose union corresponds to a spanning-tree of the chain. Two corresponding vectors $\boldsymbol{\theta}_P$ and $\boldsymbol{\theta}_Q$ represent the internal state of each branch. To derive the relationship between $\boldsymbol{\theta}_P$ and $\boldsymbol{\theta}_Q$ due to the existing constraints, we introduce an formalism to relate the state vectors to the spatial arrangement of the chain.

A. Unconstrained Chain

In order to derive the kinematics of a chain, we first derive the kinematic equations of the equivalent unconstrained system ([8], [17], [18]), where all the loops are temporary ignored and the topology of the chain is replaced by a *spanning-tree*. Such constraints are then restored in terms of kinematic relationships among moving parts.

As the structure of the kinematic equations depends on the selected spanning-tree, we exploit the symmetry of the chain and identify two independent branches $P = \{P_0, P_1, \cdots\}$ and $Q = \{Q_0, Q_1, \cdots\}$ as shown in Fig. 2, where the links $i \in \{3, 6, 9, \dots\}$ are cut apart. This results in two independent kinematic equations – one each branch – whose structure is simple to extend for a growing number of particles. Instead, the selection of a different spanning-tree results in a growing number of equations, one for each open branch in the spanning-tree [18], which might complicate the model formulation and the subsequent software implementation.

As the two branches P and Q are geometrically equivalent, we focus on the former and derive its kinematic equations. Without loss of generality, we assume the head of the chain to be fixed to an *inertial* reference frame Ω_0, as shown in Fig. 2. The following equation indicates the position of the point P_i of branch P with respect to Ω_0:

$$\boldsymbol{P}_{[i],\langle 0\rangle} \equiv \underbrace{\begin{bmatrix} \cos(\boldsymbol{e_1}^\top \boldsymbol{\theta}_P) & \cdots & \cos(\boldsymbol{e_n}^\top \boldsymbol{\theta}_P) \\ \sin(\boldsymbol{e_1}^\top \boldsymbol{\theta}_P) & \cdots & \sin(\boldsymbol{e_n}^\top \boldsymbol{\theta}_P) \end{bmatrix}}_{\boldsymbol{D}_n(\boldsymbol{\theta}_P)} \cdot \underbrace{\begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{bmatrix}}_{\boldsymbol{\delta}_n} \cdot \boldsymbol{e_i} \tag{1}$$

The state vector $\boldsymbol{\theta}_P$ contains the *joint-coordinates* of the branch P composed of n joints and $n-1$ links. For simplicity of notation, the time dependency is omitted.
The linear operator $\boldsymbol{e_j}^\top$ is a vector composed of j *ones* followed by $n-j$ *zeros*. The product $\boldsymbol{e_j}^\top \cdot \boldsymbol{\theta}$ is thus equivalent to $\sum_{i=1}^{j} \theta_i$. Each element d_i of the diagonal matrix $\boldsymbol{\delta}_n$ is the length of link i of the branch. Thereby, the dot product $\boldsymbol{D}_n(\boldsymbol{\theta}) \cdot \boldsymbol{\delta}_n$ generates the segments $\overrightarrow{P_j P_{j+1}}$, which multiplied by the vector $\boldsymbol{e_i}$ sum up yielding $\boldsymbol{P}_{[i],\langle 0\rangle} \equiv \sum_{j=1}^{i} \overrightarrow{P_j P_{j+1}}$. The notation $\boldsymbol{P}_{[i],\langle 0\rangle}$ is used to indicate the position of a point $\boldsymbol{P}_i$ with respect to the absolute frame Ω_0. A similar equation is defined for the branch Q considering the right shift of Q_0 along the X-axis of Ω_0 and its state vector $\boldsymbol{\theta}_Q$.

By splitting the chain, the constraints among the two branches P and Q are temporary removed. The following constraint equations model their mutual dependency, in terms of the state vectors $\boldsymbol{\theta}_P(t)$ and $\boldsymbol{\theta}_Q(t)$. In particular, every cut-link implies angular and spatial constraints, for which a variation of $\boldsymbol{\theta}_P(t)$ determines a variation of $\boldsymbol{\theta}_Q(t)$, and vice-versa.

B. Loop-Closure: Angular Constraints

The angular constraints require the segments $\overline{P_{i-1}P_i}$ and $\overline{Q_{i-1}Q_i}$ of the two branches of Fig. 2 to be parallel. This condition needs to be satisfied for each $i \in \{3, 6, 9, \dots\}$. The following equation defines such constraints in terms of the joint-coordinate vectors $\boldsymbol{\theta}_P$ and $\boldsymbol{\theta}_Q$, $i \in \{3, 6, 9, \dots\}$:

$$C_i^A = \sum_{j=1}^{i} \theta_{P_j} - \sum_{j=1}^{i} \theta_{Q_j} = \boldsymbol{e_i}^\top \boldsymbol{\theta_P} - \boldsymbol{e_i}^\top \boldsymbol{\theta_Q} = \boldsymbol{0} \tag{2}$$

where $\boldsymbol{e_i}$ is a n-element column vector containing i ones followed by $n - i$ zeros, which multiplied by a vector produces the sum of its first n elements.

C. Loop-Closure: Spatial Constraints

The spatial constraints implies the relative location of the joints $\boldsymbol{P}_{[i]}$ and $\boldsymbol{Q}_{[i]}$ ($i \in \{3, 6, 9, \dots\}$) to be fixed. Referring to Fig. 2, the first joints P_0 and Q_0 of the two branches are fixed to the reference frame Ω_0. In particular, P_0 coincide with the origin of Ω_0, whereas Q_0 is placed along the X-axis at a distance w.

In a similar way, the relative location of any pair of joints $\boldsymbol{P}_{[i]}$ and $\boldsymbol{Q}_{[i]}$ can be defined assuming a reference frame Ω_i attached to each link i of branch P (as shown in Fig. 2). According to the Denavit-Hartenberg convention [19]), the frame Ω_i has its origin coincident with the joint $\boldsymbol{P}_i$, the X-axis aligned to the link and pointing away from the other joint $\boldsymbol{P}_{i-1}$. For each disjoint link $\boldsymbol{P}_{[i]}$ – $\boldsymbol{Q}_{[i]}$ with $i \in \{3, 6, 9, \dots\}$, a point $\boldsymbol{Z}_{[i]}$ is defined in Ω_i to indicate the expected position of the joint $\boldsymbol{Q}_{[i]}$, when the spatial constraint is satisfied. More precisely, as $\boldsymbol{Q}_{[i]}$ is located along the Y-axis of Ω_i at a distance w, $\boldsymbol{Z}_{[i]} = [0, w]^\top$ with respect to Ω_i.

The notation $\boldsymbol{P}_{[i],\langle 0\rangle}$ earlier introduced in Eq. 1 to indicate the position of a joint $\boldsymbol{P}_i$ with respect to the absolute frame Ω_0, can be extended to the point $\boldsymbol{Z}_{[i]}$, as follows:

$$\boldsymbol{Z}_{[i],\langle 0\rangle} = \boldsymbol{P}_{[i],\langle 0\rangle} + R\left(\boldsymbol{e_i}^\top \boldsymbol{\theta}_P\right) \cdot \boldsymbol{Z}_{[i]} \tag{3}$$

where $R(\alpha) = \left[\begin{smallmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{smallmatrix}\right]$ is a rotation matrix to compensate the angular displacement between Ω_0 and Ω_i.

The following equation defines the spatial constraints condition $\boldsymbol{Z}_{[i],\langle 0\rangle} \equiv \boldsymbol{Q}_{[i],\langle 0\rangle}$, as function of the vectors $\boldsymbol{\theta}_P$ and $\boldsymbol{\theta}_Q$ for each $i \in \{3, 6, 9, \dots\}$:

$$C_i^S = \underbrace{\boldsymbol{D_n}(\boldsymbol{\theta_P})\boldsymbol{\delta}_n \boldsymbol{e_i} + R(\boldsymbol{e_i}^\top \boldsymbol{\theta_P}) \begin{bmatrix} 0 \\ w \end{bmatrix}}_{\boldsymbol{Z}_{[i],\langle 0\rangle},\ \text{where } \boldsymbol{Z}_{[i]} = [0,w]^\top} - \underbrace{\boldsymbol{D_n}(\boldsymbol{\theta_Q})\boldsymbol{\delta}_n \boldsymbol{e_i}}_{\boldsymbol{Q}_{[i],\langle 0\rangle}} = \boldsymbol{0} \tag{4}$$

In the above equation the terms $\boldsymbol{D}_n(\cdot)$ avoid redundant computation of the same trigonometric functions. The matrix $\boldsymbol{D}_n$ is indeed computed once and the result used multiple times.

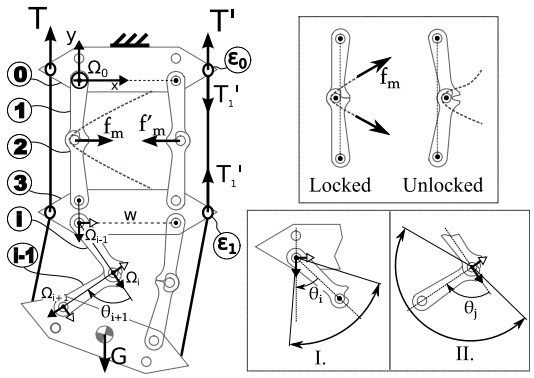


Fig. 3: Forces and reference frames.

D. Spatial Velocity

To complete the kinematic model and for later application in the dynamic model, we derive the velocity of the point $\boldsymbol{P}_{[i],\langle 0\rangle}$ and extend the result to $\boldsymbol{Z}_{[i],\langle 0\rangle}$.

Let us consider the last point $\boldsymbol{P}_{[n],\langle 0\rangle}$ of branch P. Its time derivative, $\dot{\boldsymbol{P}}_{[n],\langle 0\rangle}$, relates spatial and angular velocities $\dot{\boldsymbol{P}}_{[n],\langle 0\rangle} = \boldsymbol{J}\cdot\dot{\boldsymbol{\theta}}_{\boldsymbol{P}}$, where $\boldsymbol{J} = \frac{\partial \boldsymbol{P}_{[n],\langle 0\rangle}}{\partial \boldsymbol{\theta}_P}$ is a Jacobian matrix. By referring to Eq. 1, the t-th column of $\boldsymbol{D}_n$ contains trigonometric functions in the parameter $\boldsymbol{e_t}^\top\boldsymbol{\theta}_P = \sum_{i=1}^{t}\theta_{P_i}$. The derivative of each t-th column contributes to the s-th element $\frac{\partial \boldsymbol{P}_{[n],\langle 0\rangle}}{\partial \theta_s}$ of $\boldsymbol{J}$ only if $t \geq s$:

$$\frac{\partial \boldsymbol{P}_{[n],\langle 0\rangle}}{\partial \theta_s} = \mathbf{1}\times\boldsymbol{D_n}(\boldsymbol{\theta_P})\cdot\boldsymbol{\delta_n}\cdot diag(\boldsymbol{e_n}-\boldsymbol{e_{s-1}})\cdot\boldsymbol{e_n} \quad (5)$$

where $(\mathbf{1}\times) = \left[\begin{smallmatrix}0 & -1\\ 1 & 0\end{smallmatrix}\right]$ is the anti-symmetric exchange matrix that multiplied by $\boldsymbol{D}_n$ yields its derivative because $\mathbf{1}\times\left[\begin{smallmatrix}\cos(\cdot)\\ \sin(\cdot)\end{smallmatrix}\right] = \left[\begin{smallmatrix}-\sin(\cdot)\\ \cos(\cdot)\end{smallmatrix}\right]$; $\boldsymbol{\delta}_n$ was defined in Eq. 1; $diag(\boldsymbol{e_n}-\boldsymbol{e_{s-1}})$ is a diagonal matrix that combined with the rightmost vector $\boldsymbol{e_n}$ sums the contributions of the t-th columns of $\mathbf{1}\times\boldsymbol{D}_n\cdot\boldsymbol{\delta}_n$, where $s\leq t\leq n$.

With the above equation we can compute the derivative of $\dot{\boldsymbol{P}}_{[i],\langle 0\rangle}$ by replacing the rightmost term $\boldsymbol{e_n}$ with $\boldsymbol{e_i}$. To extend this result to compute $\dot{\boldsymbol{Z}}_{[i],\langle 0\rangle}$, we need to include the contribution of the rotation matrix $R(\cdot)$ of Eq. 3:

$$\tfrac{\partial}{\partial\theta_s}R\left((\boldsymbol{e_i}^\top-\boldsymbol{e_j}^\top)\boldsymbol{\theta}_P\right)\cdot\boldsymbol{Z}_{[i]} = \mathbf{1}\times R\left(\boldsymbol{e_i}^\top\boldsymbol{\theta}_P\right)\cdot\boldsymbol{Z}_{[i]} \quad (6)$$

$\boldsymbol{Z}_{[i]}$ is constant with respect to the reference frame Ω_i, hence independent of θ_s.

IV. Dynamics

The dynamic model defines the relationship between the accelerations and the forces existing in the system, taking into account the inertia and the mass of each component as well as the mechanical constraints. In the remainder, we refer to a state vector $\boldsymbol{\theta}$ that combines the state vectors of the branches P and Q $\boldsymbol{\theta} = \left[\begin{smallmatrix}\boldsymbol{\theta}_P\\ \boldsymbol{\theta}_Q\end{smallmatrix}\right]$. We refer to Newton-Euler equations in order to characterize the dynamics of the chain:

$$\begin{cases}\boldsymbol{F} = \boldsymbol{M}\boldsymbol{a}\\ \boldsymbol{\tau} = \boldsymbol{I}\ddot{\boldsymbol{\theta}} = \boldsymbol{I}\frac{d\dot{\boldsymbol{\theta}}}{dt}\approx\boldsymbol{I}\frac{\dot{\boldsymbol{\theta}}_k-\dot{\boldsymbol{\theta}}_{k-1}}{\Delta t}\end{cases} \quad (7)$$

The vectors $\boldsymbol{F}$ and $\boldsymbol{\tau}$ are the forces and torques acting in the system; $\boldsymbol{M}$ and $\boldsymbol{I}$ are the mass and the inertia matrices; $\boldsymbol{a}$ and $\ddot{\boldsymbol{\theta}}$ are the linear and angular acceleration vectors. It is worth noting that the second equation is in a "simple" form that only applies when the movements are constrained to two-dimensional space. As the system at hand is mainly characterized by angular velocities, we only focus on the second equation although similar conclusion can be drawn for the first equation.

We approximate the angular acceleration $\ddot{\boldsymbol{\theta}}$ with its discrete derivative as shown in Eq. 7, with k being a temporal index. The new vector $\dot{\boldsymbol{\theta}}_k$ is computed at every time step k on the basis of the current status vector ($\boldsymbol{\theta}_{k-1}$), the angular velocities ($\dot{\boldsymbol{\theta}}_{k-1}$) and the forces $\boldsymbol{\tau}$ in action. Fig. 4 shows a diagram where, starting from the initial state defined by $\boldsymbol{\theta}_{k-1}$ and $\dot{\boldsymbol{\theta}}_{k-1}$, the new state defined by $\boldsymbol{\theta}_k$ and $\dot{\boldsymbol{\theta}}_k$ is computed. All the forces in the system are transformed into an equivalent torque at the joints according to the equation:

$$\tau_j^F = \boldsymbol{F}\cdot\frac{\partial \boldsymbol{Z}_{[i]}}{\partial\theta_j} \quad (8)$$

where τ_j^F results from the application of force $\boldsymbol{F}$ to a generic point $\boldsymbol{Z}_{[i]}$ defined in Ω_i. The Jacobian $\frac{\partial \boldsymbol{Z}_{[i]}}{\partial\theta_j}$ derives from Eq. 5 and Eq. 6. Torques sum up in the vector $\boldsymbol{\tau}$.

A. Forces

Fig. 3 shows five different types of forces acting in a particle chain:

(i) Tensile forces T and T' applied to the tendons propagate through eyelets (e.g., ε_1) along the chain. For instance, the first particle of Fig. 3 is subject to a pair of equivalent opposing forces T_1' and T_1 acting at the eyelets and compressing the particle. To compute the action-line of these forces Eq. 3 applies assuming that two points $\boldsymbol{Z}_{[i],\langle 0\rangle}$ and $\boldsymbol{Z}_{[j],\langle 0\rangle}$ are coincident with the eyelets ε_i and ε_j. The vector $\boldsymbol{Z}_{[i],\langle j\rangle} = \boldsymbol{Z}_{[i],\langle 0\rangle} - \boldsymbol{Z}_{[j],\langle 0\rangle}$ indicates to the action-line of the tensile force applied to ε_i. Friction effects between the tendon and the eyelet reduce the tension at each eyelet. Referring to the propagation model we prosed in [2], the tension reduction factor corresponds to $(1-\mu|\sin(y_m)|$ where y_m is the angle enclosed by the top and bottom link of a particle as shown in Fig. 1. The tensions $T_m(t)$ and $T_m'(t)$ for the generic particle m are grouped into the vector $\boldsymbol{X}(t)$ (shown in Fig. 1) necessary to determine whether the control input $\boldsymbol{u}(t)$ ensures that $T_m(t)>0$ and $T_m'(t)>0$, such that the chain is controllable. The block "Tension model" of Fig. 4 includes two sub-models "Action-Lines" and "Tension-Propagation" described above. The block Jacob$^{\text{TEN}}$ is matrix that transforms the tensile forces applied to the eyelets into equivalent torques applied to the joints. The vector $\boldsymbol{\tau}^T$ groups these torques.

(ii) Forces f_m and f_m' exerted by the built-in actuators select the folding side by "unlocking" the particle (Fig. 3 shows the locked and unlocked conditions).

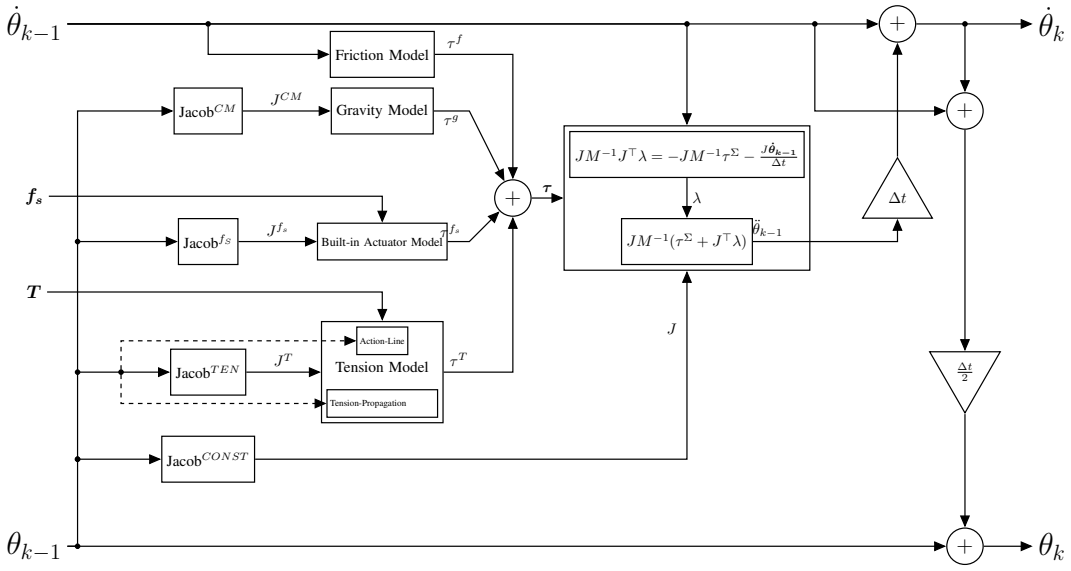


Fig. 4: Dynamic model.

A built-in actuator consists of a short wire of Shape-Memory Alloy (SMA) that contracts when heated up. As the wire can only contract 4% of its lengths, the resulting effect is spatially limited and vanishes when the link is unlocked. Given their limitation, we model f_m as a torque applied directly at the joint to unlock the particle. In the diagram of Fig. 4, a Jacobian transformation matrix Jacob^{f_m} is shown to give a complete overview of the system. Future work, will include the definition of a more accurate model also for these weak yet important forces.

(iii) Gravity force G acts at the link's center of mass associated to a point $\boldsymbol{Z}_{[i]}$ defined locally in each particle reference frame. Fig. 4 shows the block Jacob$^{\text{CM}}$ that transforms the force(s) G into equivalent torques. The vector $\boldsymbol{\tau}^G$ groups these torques.

(iv) Friction τ^f at the joints counteracts the relative rotation θ_i of two links. We model this effect as $\boldsymbol{\tau}^f = -k\dot{\boldsymbol{\theta}}$, where the constant k is empirically derived. Fig. 4 shows the block "Friction Model" whose output is the torque vector $\boldsymbol{\tau}^f$.

(v) Reaction forces $\boldsymbol{\lambda}$ at the joints maintain the two branches P and Q connected. The central block of Fig. 4 contains two sub-blocks connected by an arrow. Label of the arrow is the vector $\boldsymbol{\lambda}$. The latter is defined in form or Lagrange multipliers considering the Jacobian $\boldsymbol{J}$ of the constraints, introduced in the next subsection. The block Jacob^{CONST} relates $\boldsymbol{J}$ to $\boldsymbol{\theta}_{k-1}$.

B. Constraints Model

All constrains imposed on the system must be satisfied. In the previous section, the angular and spatial constraints are defined as two functions $C_i^A(\boldsymbol{\theta}_P, \boldsymbol{\theta}_Q) = 0$ and $C_i^S(\boldsymbol{\theta}_P, \boldsymbol{\theta}_Q) = 0$. Their time derivatives determine the relationship between the angular velocities of the constrained parts, namely how the two vectors $\boldsymbol{\theta}_P$ and $\boldsymbol{\theta}_Q$ influence each other. Let $C(\boldsymbol{\theta}) = \begin{bmatrix} C_i^A \\ C_i^S \end{bmatrix}$ be the combination of C_i^A and C_i^S for each $i \in \{3, 6, \dots\}$. The derivative needs also to be null, $\frac{dC}{dt} = \frac{\partial \boldsymbol{C}}{\partial \boldsymbol{\theta}} \cdot \dot{\boldsymbol{\theta}} = \boldsymbol{J} \cdot \dot{\boldsymbol{\theta}} = \boldsymbol{0}$, in order to satisfy the constraints. $\boldsymbol{J}$ is the Jacobian of C for which Eq. 5 applies. This combined with Eq. 7 yields:

$$J\boldsymbol{I^{-1}}\boldsymbol{\tau}\Delta t = J\dot{\boldsymbol{\theta}}_{\boldsymbol{k}} - J\dot{\boldsymbol{\theta}}_{\boldsymbol{k-1}} \tag{9}$$

This equation has two unknowns that are $\dot{\boldsymbol{\theta}}_k$ and $\boldsymbol{\tau}$. In order to have the constraints satisfied, the new angular velocity $\dot{\boldsymbol{\theta}}_k$ must satisfy the condition $\boldsymbol{J}\dot{\boldsymbol{\theta}}_{\boldsymbol{k}} \equiv \boldsymbol{0}$. Consequently, if we assume the latter condition to be true, Eq. 9 can be solved for $\boldsymbol{\tau}$ that is the sum of all the actuation and inherent forces (i.e., gravity) – which are known – and the reaction forces at the constraints – which can be computed in form of Lagrange multipliers $\boldsymbol{J}^\top\boldsymbol{\lambda}$ (only shown in the upper equation of Fig. 4). Once $\boldsymbol{\tau}$ is known, Eq. 9 can be solved for $\dot{\boldsymbol{\theta}}_k$, and the latter integrated over time to obtain the new vector $\boldsymbol{\theta}_k$.

C. Limit Constraints

Referring to the symmetrical layout of a particle (Fig. 3), we distinguish between two types of joints:

I. joints in every top/bottom link (e.g., link 0 and 4).
II. joints in every other link (e.g., links 1 and 2).

Fig. 3 shows in detail the two types I and II. Joints are designed to constrain the relative rotation of the connected links within a certain range $\theta_i \in [I_{min}, I_{max}]$ and $\theta_i \in [II_{min}, II_{max}]$, which depends on the specific joint. Whenever a force drives a link to rotate beyond the admissible

angle, according to the action-reaction principle, another reaction force immediately counteracts the first one, nullifying it. This condition introduces a non-linearity in the system that we handle by temporarily concatenating a new constraint to the Jacobian matrix introduced in Eq. 9, which forces the corresponding angular velocity to be zero.

V. Evaluation

In this section we evaluate the accuracy of the model with respect to a real prototype. A predefined sequence of actuation forces is used to control the prototype and as input to the model. The latter predicts the state of the chain at each point in time, and in particular position and inclination of the bottom link (tail of the chain). At corresponding points in time, we annotate position and inclination of the tail of the real chain (i.e., last particle) and estimate how the model-predictions diverge from reality.

A. Experimental Setup

We program a three-particle prototype to execute the folding experiment reported in Table I, where all the particles are folded starting from the bottom particle (C) and proceeding upwards to the top particle (A) (Fig. 1). At the specified point in time, the built-in actuator of particle i (i.e., SMA) unlocks one of the two flexible links, exerting the force f_s. The local actuator stays active for an average period of 0.69 s. We have measured this activation period as the shortest time to fully unlock a particle without damages to the folding chain. After a short delay from the activation of the built-in actuator, the external actuator winds up the tendons applying the force T to fold the unlocked particle and thus to bend the chain. The external actuator is active for an average time of 1.05 s. Table I reports for each particle the folding **side**, the **start** and the **end** time of the unlocking and folding phase.

With the aid of a video camera, we record the execution of the above experiment and annotate for each point in time the location and the inclination of the last particle of the chain that we consider to be the best representative of the entire chain as it is the only particle always in motion. In order to correct the small distortion resulting from an inevitable misalignment of the camera and the prototype, we pre-process the video applying a perspective transformation.

The duration of the video is less than 8 s, therefore we decide to manually annotate in each frame the location of the two lateral corners of the tail particle, which are the easiest features to identify in order to minimize the measurement error. We then calculate the position of the center of mass of the particle (mid point of the segment between the two points) and its inclination with respect to an equivalent frame of reference as we have defined in the model.

The model predicts the behaviour of the chain undergoing the same experiment, with exactly the same sequence of actions and timing (Table I). After each iteration the position of the center of mass and the inclination of the tail particle is annotated to compare against reality. Dimensions and mass of the chain and its sub-components are directly measured on the real prototype. The inertia is computed with the aid of the CAD software used to design the particle. The coefficient of friction μ between eyelets and tendons is set to 0.36, according to measurements on the real prototype. The coefficient of friction k at the joints is set to an empirical value of 1.6, since a direct measurements is not possible. The torque to unlock the particle is set to $0.47\,N\,m$, which is the maximum force f_m the local actuator (Shape-Memory Alloy) can exert. The synchronization between the real prototype and the output of the model is calculated based on the frame rate of the video.

TABLE I: Folding Experiment

Particle	Unlock Particle			Fold chain	
	Side	Start [s]	End [s]	Start [s]	End [s]
3. (C)	Right	1.45	2.14	1.94	3.00
2. (B)	Right	3.29	3.98	3.79	4.91
1. (A)	Left	5.133	5.82	5.63	6.66

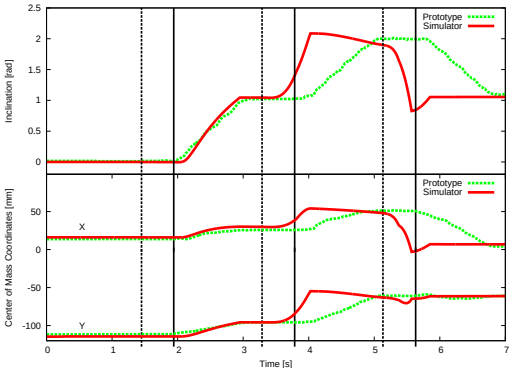


Fig. 5: Evaluation of the model against the real prototype.

B. Comparison

The two diagrams of Fig. 5 show for each point in time the accuracy of the model compared to the real prototype. The upper diagram shows the inclination of the last particle of the two systems, whereas the lower diagram reports the X and Y coordinates of its center of mass. The vertical dashed and solid lines indicate, alternately, the activation time of the built-in and external actuators.

We notice that between the first and second activations of the built-in actuator (time = 3.2 sec), the behaviour predicted by the model follows the real prototype with average absolute errors lower than 0.049 rad, 1.15 mm, 1.80 mm respectively for the angle, X, and Y curves. We must also consider, however, that even the first part of the graphs where the offset between the real and predicted values should be zero, since both chains are in their resting position, presents a deviation due to measurement errors, coordinate and perspective transformation of the video.

In the right part of the diagram, starting from $t = 3.29\,s$, which is the second activation of the built-in actuator, the

curves diverge. The discrepancy is essentially an offset in the reaction time of the two systems, whereas the final posture of the two chains is comparable after each activation.
We conjecture this behaviour is due to different actuation forces in the two systems. In the real prototype the tendons are pulled up at a constant velocity, hence the applied tensile forces vary in order to ensure this condition. Also, the force to retain a posture is minimal and depends on the posture itself.
In the simulated chain the actuation begins with a maximum peak force of $3.0\,N$ applied to the tendons right after the activation of the actuator, and then exponentially decreased to a value of $0.5\,N$. This value is the minimum force necessary to retain the posture of the chain after each actuation, any lower value makes the chain unfold – even after the first actuation. This force, combined with the torque necessary to unlock a particle, results in an early folding of the chain: during the second activation, indeed, inclination and position of the last particle change already $0.2 - 0.3\,s$ before the activation of the external actuator. A lower torque prevents the particles from unlocking.

However, as our first goal is to predict the feasibility of a target configuration upon the application of maximum actuation forces, the time offset between the simulated and the real chain is acceptable. The relevant result is that in both the simulated and the real chain, the final postures are equivalent. In case the maximum forces are not are not sufficient, the model shows a divergence in the final posture (not shown in the graph). In this way, the same model can be used in a model-predictive controller to seek the set of minimal actuation forces necessary to achieve a target configuration.

VI. Discussion of Computational Overhead

The presented model minimizes the number of trigonometric functions to compute. Considering the diagram in Fig. 4, the blocks named "Jacob" indicate a force transformation, for which a computation of a Jacobian matrix is necessary at each time step k. However, such Jacobian matrices are all based on the same $\boldsymbol{D}_n$ defined in Eq. 1. We approximate the generic Jacobian $J(\boldsymbol{\theta})$, assuming it constant for a given interval $k \rightarrow k+1$. This allows $\boldsymbol{D}_n$ to be computed only once at the beginning of each time step and then applied to derive other Jacobian matrices (Eq. 5). In addition, the matrix $\boldsymbol{D}_n$ is necessary to compute the spatial arrangement of the particles in the chain (i.e., to visualize the curvature) and to compute the "action lines" of the tensile forces extorted by the tendons. This notably reduces the computational overhead and make the model amenable for implementation on a constrained embedded computing platform.

VII. Conclusion

We have introduced the concept of shape-shifting surface, a robotic system able to display arbitrary 3D objects. Composed of modular folding robotic chains, each chain outlines a corresponding section of the target object. A chain is a concatenation of modular robotic particles, where each particle controls the local curvature. As the final resolution of the surface depends on size and number of such particles, a minimalistic design ensures their miniaturization.

However, optimal actuation strategies are necessary to also minimize the forces acting in the system, hence not to limit its scalability. Consequently, in order to enable optimal planning and model-predictive control, we derived a dynamic model of a chain to predict its behaviour upon the application of maximum actuation forces.

To asses the accuracy of the model, we compared the behaviour of a simulated chain – based on the model – against a real prototype. When both chains are subject to the same actuation sequence, the posture the simulated chain after each actuation corresponds to the posture of the real chain. Despite the reaction time of the two systems presents an offset, mainly ascribable to different intensities of the actuation forces, the model predicts the behaviour of the real prototype with acceptable accuracy.

References

[1] M. Lasagni and K. Römer, "Force-guiding particle chains for shape-shifting displays," in *IROS*, 2014.
[2] M. Lasagni and K. Römer, "Force model of a robotic particle chain for 3d displays," in *SAC*, 2015.
[3] A. Agarwal, S. Shah, S. Bandyopadhyay, and S. Saha, "Dynamics of serial kinematic chains with large number of degrees-of-freedom," *Multibody System Dynamics*, 2014.
[4] H. Chaudhary and S. K. Saha, "Constraint wrench formulation for closed-loop systems using two-level recursions," *Journal of Mechanical Design*, 2007.
[5] J. Baumgarte, "Stabilization of constraints and integrals of motion in dynamical systems," *Computer Methods in Applied Mechanics and Engineering*, 1972.
[6] S. Wittmeier, M. Jantsch, K. Dalamagkidis, M. Rickert, H. Marques, and A. Knoll, "Caliper: A universal robot simulation framework for tendon-driven robots," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, 2011.
[7] U. M. Ascher, H. Chin, L. R. Petzold, and S. Reich, "Stabilization of constrained mechanical systems with daes and invariant manifolds," *Mechanics of Structures and Machines*, 1995.
[8] R. Featherstone and D. Orin, "Robot dynamics: equations and algorithms," in *Robotics and Automation, ICRA.*, 2000.
[9] W. Schiehlen, "Multibody system dynamics: Roots and perspectives," *Multibody System Dynamics*, no. 2, pp. 149–188.
[10] R. Featherstone, "A divide-and-conquer articulated-body algorithm for parallel o(log(n)) calculation of rigid-body dynamics. part 2: Trees, loops, and accuracy," *The International Journal of Robotics Research*, 1999.
[11] S. V. Shah, S. K. Saha, and J. K. Dutt, *Dynamics of Tree-type Robotic Systems*. Springer Netherlands, 2013.
[12] P. Flores and J. Ambrsio, "Revolute joints with clearance in multibody systems," *Computers & Structures*, 2004.
[13] P. Fritzkowski and H. Kamiński, "Dynamics of a rope modeled as a multi-body system with elastic joints," *Computational Mechanics*, 2010.
[14] G. Bastos, R. Seifried, and O. Brüls, "Inverse dynamics of serial and parallel underactuated multibody systems using a dae optimal control approach," *Multibody System Dynamics*, 2013.
[15] S. Mukras, N. H. Kim, N. A. Mauntler, T. L. Schmitz, and W. G. Sawyer, "Analysis of planar multibody systems with revolute joint wear," *Wear*, 2010.
[16] J.-J. Lee and Y.-H. Lee, "Dynamic analysis of tendon driven robotic mechanisms," *Journal of Robotic Systems*, 2003.
[17] P. E. Nikravesh, *Planar Multibody Dynamics: Formulation, Programming and Applications*. CRC Press, Inc., 2007.
[18] S. Shah, S. Saha, and J. Dutt, "Dynamics of robotic systems," in *Dynamics of Tree-Type Robotic Systems*, 2013.
[19] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices." *Trans. of the ASME. Journal of Applied Mechanics*, 1955.

Paper D

DOI: 10.1115/SMASIS2016-9263
Paper No. SMASIS2016-9263
Link:http://proceedings.asmedigitalcollection.asme.org
ISBN: 978-0-7918-5049-7

Proceedings of the ASME 2016 Conference on Smart Materials, Adaptive Structures and Intelligent Systems
SMASIS2016
September 28-30, 2016, Stowe, VT, USA

SMASIS2016-9263

DYNAMIC MODEL OF TENDON-DRIVEN ROBOTIC CHAINS FORMING A SHAPE-SHIFTING SURFACE

Matteo Lasagni
Institute for Technical Informatics
Graz University of Technology
Graz, Austria
Email: lasagni@tugraz.at

Kay Römer
Institute for Technical Informatics
Graz University of Technology
Graz, Austria
Email: roemer@tugraz.at

ABSTRACT

This paper presents Kinematics and Dynamics of a Shape-Shifting Surface, a robotic system able to take on the shape of arbitrary connected 3D surfaces. Such a surface, which we introduced and described in previous work, consists of piecewise controllable chains in turn composed of serially connected foldable "robotic particles". Aiming at a high resolution rendering, where tiny particles need to be combined in a large number, a tendon-driven design is a lightweight and scalable solution.

However, improper actuation strategies might expose the system to undesired forces, which can compromise its integrity and stability. To tackle this problem, optimal actuation and planning strategies are required to anticipate unacceptable situations. To this end, a dynamic model is derived to predict the reaction of the system subject to control actions. Being the system both tendon-driven and under-actuated, we have to overcome a number of challenges in deriving this model.

NOMENCLATURE

$\boldsymbol{F}_n$ Force due to friction between tendon and eyelet
$\boldsymbol{J}_c$ Jacobian matrix
L_n, R_n Force to unlock the left/right side of particle n
$\dot{r}$ Folding velocity
$\boldsymbol{s}_i$ Versor associated to link i
$\hat{\boldsymbol{s}}_i$ 90° counter-clockwise rotation of the versor $\boldsymbol{s}_i$
$\boldsymbol{T}$ Tensile force applied to a chain
$\boldsymbol{T}_n$ Tensile force applied to particle n
$\boldsymbol{v}$ Linear velocity
ε Position of an eyelet
θ_i Relative joint coordinate
$\boldsymbol{\theta} = [\theta_i]$ Internal state vector of a particle
τ Torque
ϕ^c Kinematic constraints
Ω_n Local frame of reference of particle n
ω Angular velocity

INTRODUCTION

A Shape-Shifting Surface is a *robotic adaptive structure* devised to dynamically take on the shape of arbitrary connected 3D surfaces. It consists of parallel foldable chains conceived to approximate the contour of corresponding slices of the target object. A chain is in turn a concatenation of modular "robotic particles", which result in a piecewise controllable structure. Each particle is responsible for the local curvature of the chain.

Among the potential applications, we envision the creation of shape-shifting displays to enrich traditional computer applications by endowing users with tangible experience. A shape-shifting display can improve the communication among different domain experts (e.g., scientist and investors) and thus facilitate the explanation of complicated concepts. Also, our daily experience can benefit of new tools like "instant prototyping" and 3D-fax [1]. Furthermore, shape-shifting surfaces can enable "programmable molds" to reduce the gap between "mass production" and "customized products", which can eventually coexist in

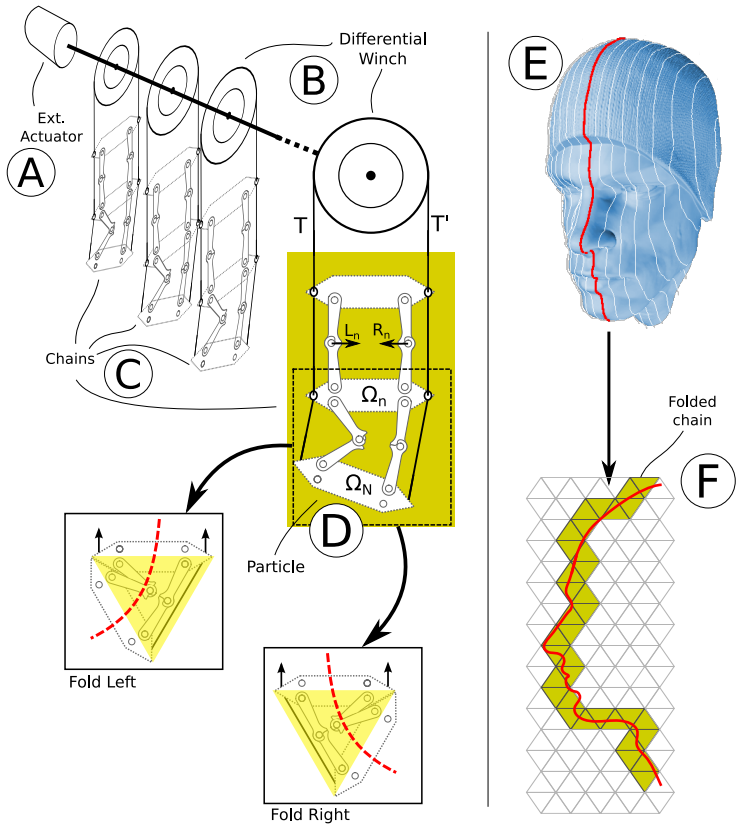


FIGURE 1. Overview of the system and its modular components.

the same manufacturing process. This would enable innovative production lines that instantaneously change their configuration to allow *high* volume production of *highly* personalized products.

In the broader vision of to emergent fields "Programmable Matter" [2, 3] and its specialization "Shape-Shifting Materials", objects having mutable physical characteristics can be based on modular robots that becomes the "programmable particles" of the material. The physical characteristics depend on the configuration of such robots (e.g., color) and on their reciprocal arrangement (which define the overall shape). With particular focus on Shape-Shifting Materials, different solutions have been proposed so far, ranging from those aiming to maximize the freedom of movement of the robotic particles [4, 5] – thus enabling the formation of any arbitrarily complex geometry, in spite of a significant mechanical complexity – to those aiming to minimize the mechanical complexity of the robotic particles – in order to simplify their manufacturing and scalability, though limiting the formation of arbitrary shapes [6].

Between these two extremes, in [7] we presented a prototype of a Shape-Shifting Surface, composed of multiple parallel robotic chains, whose curvature can change in order to outline a slice of a target 3D object.
As the resolution of the display is key to widen the range of possible applications, the mechanical design plays a relevant role to eventually make particles amenable for miniaturization. Complex mechanisms would require sophisticated technologies which make it difficult, if not prohibitive, to realize the system at low cost. In addition, as the size of particles reduces, the number of particles forming a surface grows in order to maintain the overall size constant. Scalability becomes another key aspect to drive the mechanical design already from the initial stage.

Our design, summarized in Fig. 1, consists of multiple piecewise foldable chains (C), composed of sequentially connected particles (D). Each robotic particle controls the curvature of the chain by means of a built-in mechanism that selects the folding side. Fig. 1 shows the two possible layouts of a particle folded

either to the left or to the right. In both cases, the particle outlines an equilateral triangle.
If particles were designed with built-in actuators, an extension of the chain would require to scale up the actuation forces, with consequent increase of the size and weight of the actuators and thus of the weight of the chain itself. In this way, the scalability would be compromised. To overcome this problem, our solution relies on a tendon-driven approach, where particles are remotely actuated by an external actuator (A). A pair of tendons deployed alongside each chain exerts forces compressing the particles, which eventually fold. Once all the particles in the chain are folded, to retain the posture of the latter, it is sufficient to hold a constant tension on the tendons. This simplifies the design even further as no latches are needed.
A set of differential-gear-winches (B) decouples the relative motion of the two tendons actuating a chain and also decouples the actuation among chains. In this manner, an extension or addition of chains only requires an upgrade of the external actuator, without any impact on the size of particles and their scalability.

Particles are constrained to fold in the 2D space, thus chains can outline the contour of a 2D shape. As fully folded particles outline an equilateral triangle, a chain whose particles are all folded always lies on a triangular grid. This makes it straightforward to determine the particle configuration (i.e., the folding side) given a specific target slice. As the contour of the target shape overlayed on the triangular grid (F) intersects a unique sequence of triangles, where each one corresponds to a particle in the chain, the relative position of consecutive triangles indicates the folding side of the particle . In this manner, it is straightforward to identify the final configuration of the whole chain [7].

Problem Statement Despite the adopted solution solves the scalability problem from a mechanical perspective, the actuation of the system presents some challenges. Focussing on a single chain, proper actuation strategies must guarantee the integrity and the correct functioning of the whole system.

While it is straightforward to infer the *final* configuration of a chain (i.e., the folding side of its particles) given a target shape to approximate, multiple *intermediate* configurations are necessary to progressively reach the final configuration starting from an initial one. Indeed, assuming many particles per chain, it is not reasonable to assume that particles can all actuate simultaneously. In the prototype we described in [7], for example, the electrical current necessary to unlock one particle (i.e., select its folding side) is about 400 mA, which makes it prohibitive to operate all particles at the same time. Also, for particularly elaborated shapes it might be explicitly necessary to proceed gradually towards the final configuration in order to avoid undesired deadlocks and the occurrence of overwhelming forces.

Fig. 2 depicts the folding process of a chain, which includes multiple intermediate configurations to reach the final one. If not appropriately planned ahead, the intermediate configurations can lead to intense forces (e.g., due to leverage effects), even higher than those required to retain the final configuration. At each intermediate configuration, a set of unfolded particles needs to be selected. For example, as shown in Fig. 2, the intermediate configuration shows two particles (in white) yet to be folded before outlining the target "head profile". To avoid inconvenient selections, resulting in undesired leverage effects, optimal planning strategies (see Fig. 2) are needed to indicate which particle(s) to fold at the intermediate configuration *i*.

To predict the behavior of the system and thus plan ahead optimal actuation strategies, we derive the kinematic and the dynamic model of the system with particular focus on the robotic particle. This also enables model-predictive control to operate the system, which is necessary to compensate for the total absence of sensors in the system, in order to limit its cost and complexity. To this end, the behavior of the system is predicted upon the application of control forces; the latter are iteratively adjusted in order to achieve the expected behavior (i.e., configuration). As we proposed in [8], this method can be used to identify a set of *minimal actuation forces*.

To the best of our knowledge, existing techniques to model open-chain, closed-chain and, more in general, tree-type multibody systems [9–14], do not support compliant elements (i.e., tendons) but mainly address the case with only rigid bodies. An exception can be found in [15], where Lee et al. propose a method specific for modeling tendon driven-robots. However, the presence of pulleys to guide the tendons, makes their system equivalent to a rigid body system, whose links are remotely actuated. In our case, only two tendons drive all the particles in the whole chain. This constitutes a challenging situation as the effect of tendons on particles' layout is difficult to predict, especially because particles result to be *under-actuated* systems with three-degrees of freedom.

In the remainder of the paper, a system overview is provided to clearly understand the overall design and the principal components; the modeling methodology indicates how the subsequent kinematic and dynamic models are formulated for each sub-component and how the resulting models can eventually be combined together.

1 System Overview and Modeling Methodology

This section provides a general overview of the system and its sub-components. As the Shape-Shifting Surface presents a modular design, an efficient methodology to formulate kinematic and dynamic models consists in analyzing each elementary subcomponent individually. This section presents the hierarchical structure of a Shape-Shifting Surface and identifies the basic subcomponents and their mechanical "coupling". This allows us, for example, to model a particle as an independent system, whose outputs are the effects the particle produces on the other components and whose inputs are the effects the particle is subject to.

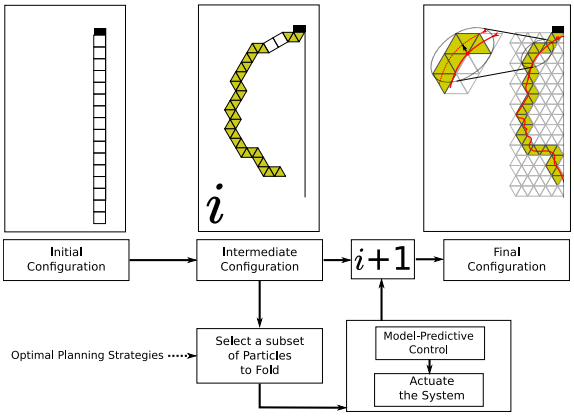


FIGURE 2. Actuation, Planning and Control of a chain: from the initial to the final configuration.

1.1 System Design

A lightweight solution to guarantee high scalability and enhanced resolution of the shape-shifting surface consists in multiple *tendon-driven* robotic chains each one responsible for outlining a slice of the 3D model to be rendered. Considering that a dedicated actuator per chain would undermine scalability and also raise costs and complexity, one single actuator (Motor in Fig. 3) is designated to actuate all the chains simultaneously. However, because each chain outlines a specific slice of the target 3D model, the final configuration of two randomly selected chains is unlikely the same. This calls for a mechanism to decouple actuation of multiple chains, as if they were individually actuated by dedicated actuators. This is possible by means of differential gears between the main shaft of the motor and the spools winding up the tendons.

Fig. 3 shows the principal components of the system and the design of a chain. A chain consists of sequentially connected robotic particles, each able to control its local curvature. A pair of tendons deployed alongside the chain exert tensile forces (i.e., T and T') necessary to fold particles and thus bend the chain as required. A particle consists of six linked bars that outline a rectangle. The two lateral sides of the rectangle, resulting from two adjoining links, can be controlled to transform the particle into an equilateral triangle as previously explained and shown in Fig. 1. The lateral sides are designed to "lock" the particle in the initial rectangular shape. When a small force (see forces L_n and R_n in Fig. 3) pulls the mid-joint towards the center of the particle, the side unlocks and hence the particle folds as soon as the external tensile forces T and T' compress the particle.

Assuming for simplicity that one particle is folding on the right side, only the right tendon winds up while the other slightly unwinds. As it is not convenient to individually actuate the two tendons, which would required two dedicated actuators per chain, our solution relies on a differential-winding mechanism that balances the tensile forces between the two tendons: the external actuator (motor) drives a differential-gear (Fig. 3) connected to two winches that in turn wind up the tendons. In case of different winding speed between the two tendons, the differential-mechanism automatically compensate for that.

This mechanism balances the forces between tendons of the same chain, but also decouples the actuation among chains, allowing one single external actuator to operate the whole system. The only constraint is that chains need a synchronous and uniform coordination: if a chain actuates and folds n particles also all the other chains must fold exactly the same number n of particles. This defines a control challenge for which optimization and ahead planning are required not only for a single chain. However, as eventually all the particles of the system need to be fully folded, it is reasonable to assume that a feasible actuation strategy always exists. A model of the system to predict its behavior before actuation is derived in the remainder of the paper.

1.2 Modeling Methodology

The hierarchical structure of the system, where the shape-shifting surface consists of multiple foldable chains, in turn composed of sequentially connected robotic particles, allows us to systematically model the system by analyzing each sub-component separately and then eventually combine the resulting sub-models to obtain the overall model of the complete system. The first step is therefore to understand how sub-components influence each other, in order to define a consistent set of inputs and outputs of each sub-model that allows their subsequent combination. With a different approach, in [8] we derived the dynamic model of a single chain, where the chain was modeled without considering its modularity, as we instead do in this work in order to obtain a set of simpler equations.

Inspired by the concept of "Power-Oriented Graph" [16], we apply this technique to model the shape-shifting surface, as shown in Fig. 3. The power flowing among physical systems is depicted by means of a pair of opposite arrows, each one representing the physical quantity responsible for the power transfer.

The mechanical power that the motor (external actuator) exchanges with the chains connected to its shaft, transfers in form of an angular velocity ω from the external actuator to the chains and returns back to the actuator in form of a torque τ, namely the reaction of the chains. The inner product $\tau \cdot \omega$ corresponds to the instant power being transferred from the external actuator to the driven chains. For example, assuming a constant angular velocity ω, an increment of the load provokes an increment of the reaction torque τ and thus of the total power supplied.

The angular velocity ω applied to the differential-winch of a chain, transfers through the latter to the chain in form of winding velocities $\dot{r}$ and $\dot{r}'$. A simpler way to understand the behavior of

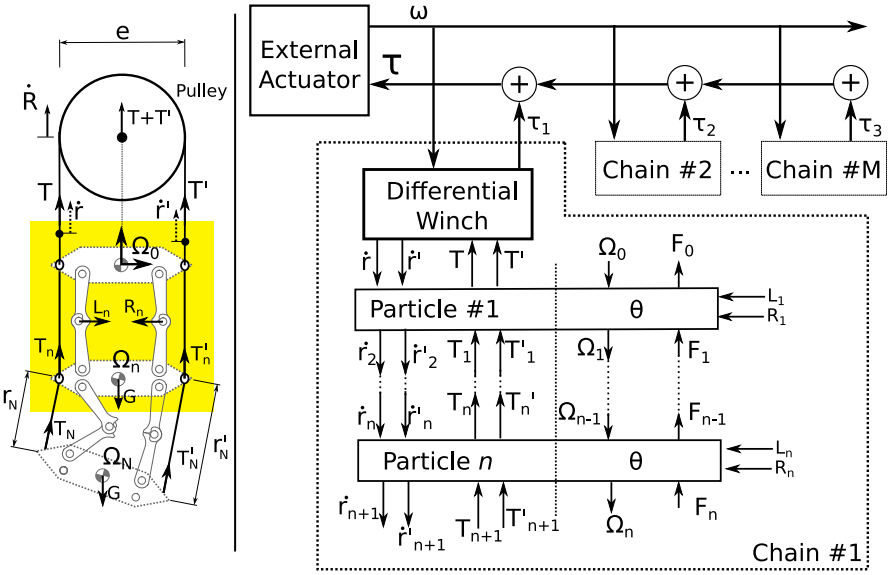


FIGURE 3. Components of the System and Power Transfer Diagram

the differential-winch is to consider an equivalent system based on a pulley that moves vertically away from the chain. Fig. 3 show this concept where the sum $T+T'$ of the tensile forces applied to the shaft of the pulley is the effect of the torque τ exerted by the motor ($(T+T')e = 2\tau$, where e is the external diameter of the winch). Similarly, the winding velocity ω is replaced by an equivalent linear velocity $\dot{R}$ that describes the vertical motion of the pulley ($2\dot{R} = e \cdot \omega$).

The tensions T and T' applied to the lateral tendons are balanced by means of the differential-winch (i.e., pulley). The latter modifies the linear velocities $\dot{r}$ and $\dot{r}'$ of the two tendons in order to maintain their tension T an T' equivalent. As the tendons are tied to the last particle of the chain, their velocity $\dot{r}$ and $\dot{r}'$ propagates through the chain. We can assume that the tendon is not extensible, therefore a movement of one extreme corresponds to an equivalent movement of the opposite extreme. When the chain is actuated at a constant velocity, the particle n reacts with a variation of the local tensile forces T_n and T'_n. These, combined with the reaction forces of all the other particles, cause a variation of the tensile forces T and T'. The differential gear converts T and T' back into a corresponding torque τ_i, which then back-propagates to the external actuator. Despite all the chains are actuated at the same angular velocity ω, because they are mechanically connected to the same shaft, the τ_i are different and depend on the particular intermediate configuration of chain i. The sum $\tau = \sum \tau_i$ is the reaction of the chains to the angular velocity ω.

Particle Dyanamics

To fully characterize the forces T_n and T'_n, also the forces L_n and R_n and the dynamics of the particle need to be taken into account. As long as the particle N is locked ($L_n = R_n = 0$), the local velocities $\dot{r}_N$ and $\dot{r}'_N$ are null regardless of the state of the external actuator. Assuming instead that particle N is unlocked on the left side ($L_n > 0$), a variation of the velocity $\dot{r}_N$ (i.e., the acceleration $\ddot{r}_N$) resulting from a variation $\ddot{r}$, causes the force T_N, depending on the dynamic state of the bottom link and according to Newton's second law. The dynamics of the generic particle n are a function of its internal state $\boldsymbol{\theta}$, but are also influenced by the other particles in the chain. For example, the weights of the particles following the head particle "#1", result in a force applied to the bottom link of particle "#1" that impairs its movement. Similarly, an acceleration of the bottom link of particle "#1" has an effect on all the other particles in the chain. This situation is depicted in the diagram of Fig. 3, where the inputs $\boldsymbol{\Omega}_n = [\Omega_n, \dot{\Omega}_n, \ddot{\Omega}_n]$ and $\boldsymbol{F}_n = [F_n, \tau_n]$ indicate the reciprocal influence of adjoining particles. The element Ω_n is the absolute position of the local frame of reference of particle n. The latter coincides with the center of mass of the lower link of particle $n-1$. The force F_n and the torque τ_n are the forces the tail of the chain exerts on the bottom link of particle n.

1.3 Kinematic Model

In this section, we derive the kinematic model of the system according to the consideration made in the previous section,

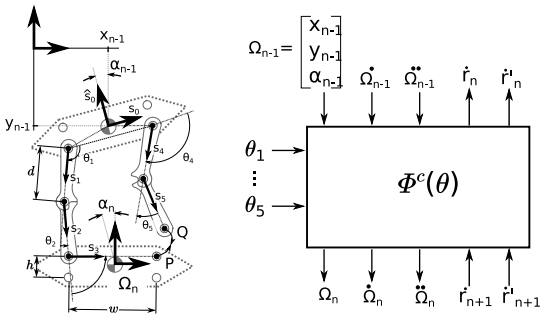


FIGURE 4. Kinematic representation of a particle without tendons: input, output and internal state.

with particular focus on the robotic particle. We formalize the relationship between its inputs, its outputs and the internal state.

1.4 Particle: Input, Output, and Internal State

Fig. 4 extends the concept introduced in the previous section and depicts all the inputs and outputs relevant to describe the kinematics of a particle. The state variables $\boldsymbol{\theta} = [\theta_i]$ (shown here as an input) and the kinematic constraints $\Phi^c(\boldsymbol{\theta})$ are shown.

Except for the head particle "#1" fixed to an inertial frame of reference, all the other particles need to be modeled as floating elements. A local reference frame Ω_{n-1} is considered for particle n as shown in Fig. 4. The abscissa is the versor $\boldsymbol{s}_0$, while the corresponding ordinate is indicated with the notation $\hat{\boldsymbol{s}}_0$.
More specifically, $\Omega_{n-1} = \begin{bmatrix} x_{n-1} & x_{n-1} & \alpha_{n-1} \end{bmatrix}^\top$ indicates the position $\begin{bmatrix} x_{n-1} & x_{n-1} \end{bmatrix}^\top$ and orientation α_{n-1} of the local reference frame of particle n with respect to an inertial reference frame (e.g., Ω_0). The relative position of the lower link Ω_n depends on the internal state of the particle, namely $\Omega_n = f(\boldsymbol{\theta})$.

To represent the internal state $\boldsymbol{\theta} = [\theta_i]$ of a particle, a number of variables equivalent to the number of degrees-of-freedom of the particle is needed. The particle, composed of six links forming a loop, presents three degrees-of-freedom. Therefore, the three joint-angles $\theta_{1,2,3}$ are sufficient to fully characterize the state of the particle, hence the position of Ω_n (to model the loop-closure constraint, we later need to also consider two additional joint-angles $\theta_{4,5}$) with respect to the inertial frame Ω_0.
Let us assume the versor $\boldsymbol{s}_i$ corresponds to the link i as depicted in Fig. 4. As $\theta_{1,2,3}$ are the relative displacements between consecutive links, the orientation of the versor $\boldsymbol{s}_i$ is equivalent to the sum of all the θ_j, with $0 < j \leq i$. Therefore, $\boldsymbol{s}_j = \left[\cos(\sum_{i=0}^{j}\theta_i), \sin((\sum_{i=0}^{j}\theta_i)\right]^\top$. With this notation, the position of Ω_n with respect to the absolute frame of reference, results to be:

$$\Omega_n = \begin{bmatrix} x_{n-1} \\ y_{n-1} \\ \alpha_{n-1} \end{bmatrix} + \begin{bmatrix} -\frac{w}{2}\boldsymbol{s}_0 - \frac{h}{2}\hat{\boldsymbol{s}}_0 + d\boldsymbol{s}_1 + d\boldsymbol{s}_2 + \frac{w}{2}\boldsymbol{s}_3 - \frac{h}{2}\hat{\boldsymbol{s}}_3 \\ \theta_1 + \theta_2 + \theta_3 \end{bmatrix} \quad (1)$$

where w and h are the dimension of the lower link of a particle; d is the length of the other links. Each versor $\hat{\boldsymbol{s}}_i$ is the derivative of $\boldsymbol{s}_i$, which geometrically corresponds to a 90° counter-clockwise rotation of the versor itself.

$$\frac{\partial \boldsymbol{s}_i}{\partial x} = \frac{\partial \left[\cos(x), \sin(x)\right]^\top}{\partial x} = \left[-\sin(x), \cos(x)\right]^\top = \hat{\boldsymbol{s}}_i$$

1.5 Particle Loop-Constraints

So far we have considered three variables $\theta_{1,2,3}$ to define the state of a particle, corresponding to the angles enclosed by three consecutive links. With only these three variables the layout of a particle might not correspond to the layout allowed by the loop-closure of its links. For example, the joints P and Q shown in Fig. 4 might drift apart if the angles $\theta_{1,2,3}$ are not properly constrained.

To formalize this constraint, we need to introduce two additional variables θ_4 and θ_5 and derive the kinematic constraint that limits the movement of P with respect to Q. With a similar notation as before, we first represent the points P and Q as functions of $\theta_{1,2,3}$ and $\theta_{4,5}$ and then impose the constraint P ≡ Q:

$$\begin{aligned} P &\equiv -\frac{w}{2}\boldsymbol{s}_0 - \frac{h}{2}\hat{\boldsymbol{s}}_0 + d\boldsymbol{s}_1 + d\boldsymbol{s}_2 + w\boldsymbol{s}_3 \\ Q &\equiv \frac{w}{2}\boldsymbol{s}_0 - \frac{h}{2}\hat{\boldsymbol{s}}_0 + d\boldsymbol{s}_4 + d\boldsymbol{s}_5 \end{aligned} \quad (2)$$

$$\phi^c(\boldsymbol{\theta}) = P - Q = w(-\boldsymbol{s}_0 + \boldsymbol{s}_3) + d(\boldsymbol{s}_1 + \boldsymbol{s}_2 - \boldsymbol{s}_4 - \boldsymbol{s}_5) = \boldsymbol{0}$$

The above equation defines the constraints $\phi^c(\boldsymbol{\theta})$ among $\theta_{1,2,3}$ and $\theta_{4,5}$, such that P and Q are coincident. As this equation is in an implicit form, it is not directly possible to determine how a variation of a joint-angle θ_i affects all other joint-angles. A more useful notation, later necessary to build the dynamic model, requires the computation of the time derivative $\dot{\phi}^c = \frac{\partial \phi^c}{\partial \boldsymbol{\theta}} \cdot \dot{\boldsymbol{\theta}} = 0$, also null in order to have the constraint satisfied. The Jacobian $\frac{\partial \phi^c}{\partial \boldsymbol{\theta}}$ defines a linear relationship among the variation of the angles θ_i, namely their velocities $\dot{\theta}_i$:

$$\frac{\partial \boldsymbol{\phi^c}}{\partial \boldsymbol{\theta}} = \begin{bmatrix} d(\hat{\boldsymbol{s}}_1 + \hat{\boldsymbol{s}}_2) + w\hat{\boldsymbol{s}}_3 \\ d\hat{\boldsymbol{s}}_2 + w\hat{\boldsymbol{s}}_3 \\ w\hat{\boldsymbol{s}}_3 \\ -d(\hat{\boldsymbol{s}}_4 + \hat{\boldsymbol{s}}_5) \\ -d\hat{\boldsymbol{s}}_5 \end{bmatrix}^\top \quad (3)$$

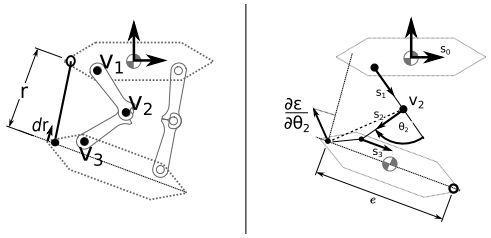


FIGURE 5. Relative velocity of tendon and eyelet and its influence on the joint angles.

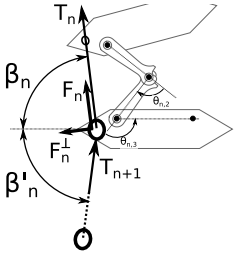


FIGURE 6. Tension propagation.

where $\boldsymbol{s}_0$ disappears because it is not dependent on any θ_i.

1.6 Particle Velocity Constraint

The interaction between tendons and eyelets alters the particle layout, hence the angular displacement of its joints. To infer how a small movement of the tendon affects the particle layout, we consider the lower link of the tail particle of the chain. As the tendons are firmly tied to it, this defines the simplest case to analyze. Similar considerations can be applied to any other particle as long as the forces acting on the eyelet due to the friction between the latter and the tendon are known. For this, we refer to the tension propagation model presented in the next section.

By referring to Fig. 5, we consider the effect due to the left tendon, tight between the last and the second to last eyelets. These are initially separated by the distance r. The tendon shortens (i.e., is pulled up) by a length dr. Under the assumption that no other force is acting in the system, the eyelet is forced to move along the path the tendon outlines by exactly the same distance dr. This movement results in a variation of the angles $\theta_{1,2,3}$ that can be computed as an inverse kinematic problem. This requires to formulate the position of the eyelet ε as a function of the angles $\theta_{1,2,3}$; then the Jacobian $\frac{\partial \varepsilon}{\partial \theta}$ indicates the relationship between a small variation of θ_i and a variation of ε:

$$\varepsilon = -\frac{w}{2}\boldsymbol{s}_0 - \frac{h}{2}\hat{\boldsymbol{s}}_0 + d\boldsymbol{s}_1 + d\boldsymbol{s}_2 - \frac{w-e}{2}\boldsymbol{s}_3 - \frac{h}{2}\hat{\boldsymbol{s}}_3 \qquad (4)$$

The i-th element of the Jacobian $J_{\varepsilon,i} = \frac{\partial \varepsilon}{\partial \theta_i}$ indicates a displacement of ε. For example, the second element $\frac{\partial \varepsilon}{\partial \theta_2}$, depicted in Fig. 5, is a vector perpendicular to the segment joining the vertex v_2 and the eyelet, which represents how the angular displacement $d\theta_i$ results in the spatial displacement $d\varepsilon$, namely $d\varepsilon = \frac{\partial \varepsilon}{\partial \theta_2} d\theta_2$. This linear relationship can be inverted in order to infer $d\theta_2$ such that $d\varepsilon = dr$, namely the eyelet moves along the tendon.
The above consideration was drawn assuming that no other forces are acting in the system. In a more realistic situation, the effect of other forces acting in the system is also considered according to the superposition principle, thus without limiting the generality of the above result.

For any other particle along the chain, the relative velocity between a tendon and an eyelet defines how the bottom link of a particle (eyelet) moves with respect to the tendon. There are three possible cases:

1. The friction between tendon and eyelet is sufficient to counteract all the other forces acting on the eyelet and thus the latter moves steadily with the tendon, and thus the velocity of the eyelet $\dot{\varepsilon} = \dot{r}$.
2. The friction is not sufficient to make the tendon drag the eyelet, therefore the latter slides along the tendon, thus the eyelet moves with a velocity $\dot{\varepsilon} < \dot{r}$.
3. The eyelet moves with a velocity $\dot{\varepsilon} > \dot{r}$ or $\dot{r} < 0$. This occurs whenever the tendon gets loose and thus the system becomes uncontrollable.

The last case does not need any particular model as the eyelet is moving unconstrained. In the first two cases, instead, the movement of the eyelet is constrained to move along the tendon according to the model earlier derived for the tail particle. In particular, in the second case, we need to estimate velocity $\dot{\varepsilon}$ as a function of $\dot{r}$. This is possible considering the friction effect and the tension propagation model presented in the following section.

2 Dynamic Model

The dynamic model is derived following established formulation [9, 17]. Focusing first on a particle, we later extend the model to the entire chain. The dynamics of the six links forming a particle must satisfy the Newton-Euler equation:

$$M\dot{v} = F^e + J_c^\top \lambda \qquad (5)$$

where M is the mass-inertia matrix of the links with respect to their center of mass; F^e are the forces acting on a particle, including the tensile forces T and T', the reaction forces due to the next particles in the chain, and the gravitational forces; $J_c^\top \lambda$ represents the forces due to the loop-closure constraint.

The velocity v and its derivative $\dot{v}$ are velocity and acceleration of the center of mass of each link composing a particle with respect to an inertial frame. They can be obtained as time derivatives of a set of equations similar to Eq. 1, defined for each link of the particle. The center of mass of a short link (i.e., the links associated to $\boldsymbol{s}_0, \boldsymbol{s}_1, \boldsymbol{s}_4, \boldsymbol{s}_5$) corresponds to the midpoint of the segment joining the two joints. It is possible to derive a relationship in the form:

$$\dot{\boldsymbol{v}} = \begin{bmatrix} \dot{\Omega}_1 \\ \dot{\Omega}_2 \\ \dot{\Omega}_n \equiv \dot{\Omega}_3 \\ \dot{\Omega}_4 \\ \dot{\Omega}_5 \end{bmatrix} = \boldsymbol{B} \begin{bmatrix} \dot{x}_{n-1} \\ \dot{y}_{n-1} \\ \dot{\alpha}_{n-1} \\ \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \end{bmatrix} = \boldsymbol{B} \cdot \dot{\boldsymbol{h}} \tag{6}$$

where the vector $\boldsymbol{h}$ combines all the inputs of a particle n: $\dot{\Omega}_n$ and its internal state. The dot product $\boldsymbol{B} \cdot \dot{\boldsymbol{h}}$ results in a vector that indicates for each link of a particle the *linear* velocity of the center of mass with respect to an inertial frame of reference.

2.1 Particle Loop-Constraints

The constraint $\phi^c = \boldsymbol{0}$ formulated earlier in the kinematic model, which enforces the loop closure of the particle, must be satisfied also in Eqn. 5. The last term of the latter equation, $\boldsymbol{J}_c^\top \boldsymbol{\lambda}$, represents the forces (w.r.t an inertial frame of reference) necessary to maintain the six links of a particle connected. More precisely, it is the force that makes P and Q (see Fig. 4) coincide.

The matrix $\boldsymbol{J}_c$ needs to be built as a relationship among the elements of the vector $\dot{\boldsymbol{v}}$, in order to be consistent with Eq. 5.

$$\boldsymbol{J}_c \cdot \dot{\boldsymbol{v}} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \begin{bmatrix} \boldsymbol{I} \\ \frac{\hat{\boldsymbol{s}_3} w + \boldsymbol{s}_3 h}{2} \end{bmatrix} \\ \boldsymbol{0} \\ -\begin{bmatrix} \boldsymbol{I} \\ \frac{\hat{\boldsymbol{s}_5} d}{2} \end{bmatrix} \end{bmatrix} \cdot \dot{\boldsymbol{v}} = \boldsymbol{0} \tag{7}$$

If we combine Eq. 7 and Eq. 6, the inner product $\boldsymbol{J}_c \cdot \boldsymbol{B}$ contains the Jacobian $\frac{\partial \phi^c}{\partial \boldsymbol{\theta}}$ earlier derived in Eq. 3.

While the last term λ is generally computed in form of Lagrange multipliers [17], after a re-arrangement of Eq. 5, a more convenient solution can be found by setting $\lambda = \overline{PQ}$, which is the vector connecting P to Q. As this leads to an acceptable stability of the constraints (subsequently reported in the evaluation), other stabilization techniques [18] are not required.

2.2 Friction Model

The friction between eyelets and tendons determines how the tension of the latter propagates along the chain.

The tension $\boldsymbol{T}_n$ of a tendon traversing particle n decreases by an amount $\boldsymbol{F}_n$, due to friction between tendon and eyelet; the residual force $\boldsymbol{T}_{n+1} = \boldsymbol{T}_n - \boldsymbol{F}_n$ propagates to the next particle $n+1$. In a chain of N particles, the force F_N acting on the tail particle N, where the tendons are tied to, is always equivalent to the tension of the tendon $\boldsymbol{F}_N \equiv \boldsymbol{T}_N$. In case the force $\boldsymbol{F}_n$ is positive and sufficiently high to overcome all the other forces applied to the eyelet, the latter moves along with the tendon. Instead, if $\boldsymbol{F}_n$ is lower than the other forces the tendon slides through the eyelet.

Referring to Fig. 6, the force $\boldsymbol{F}_n = \boldsymbol{T}_{n+1} - \boldsymbol{T}_n$ can be computed knowing the friction coefficient μ between eyelet and tendon and the deflection angle $\beta_n + \beta'_n$. The latter derives from Eq. 4. To estimate $\boldsymbol{F}_n$, we apply the Coulomb model of friction. Assuming the eyelet can be approximated by a pin-hole, the force $\boldsymbol{F}_n$ is proportional by a friction coefficient μ to the tangential force $\boldsymbol{F}_n^\perp$, shown in Fig. 6. The tangential force $\boldsymbol{F}_n^\perp$ depends on $\boldsymbol{T}_n$ and the angle $\beta_n + \beta'_n$ as $F_n^\perp = |\boldsymbol{T}_n| \cdot \cos((\beta_n + \beta'_n)/2)$. Instead, the coefficient μ is determined after a series of empirical measurements on a real particle, to be in the range $[0.3, 0.6]$.

3 Evaluation

This section presents the results obtained with a simulator based on the formulated model. The simulator is developed in Phyton and requires the scientific packages NumPy and SciPy. The evaluation concerns:

1. The tension propagation in a folding chain.
2. The stability of the loop-constraints, hence the drift-off of the points P and Q (Fig. 4).

3.1 Tension Propagation

The dynamic model presented in the previous section is applied to predict the behavior of a three-particle chain. An actuation process is simulated to estimate how the tension propagates along the chain. The tail particle is unlocked and a tension $T = T'$ applied to the tendons. We execute two experiments:

1. The tensions T and T' applied to the tendons are constant;
2. The winding velocity of the tendons is constant, therefore T adapts to satisfy the latter condition.

We evaluate the tensions T_2 and T_3 (tensions propagate across an eyelet) on both sides of the chain and compare them with the tension applied to the chain T. The latter corresponds to the tension applied to the particle head of the chain $T = T_1 = T'_1$. As the tension propagation also depends on the posture of the chain, because friction varies with the deflection of the tendon, we also

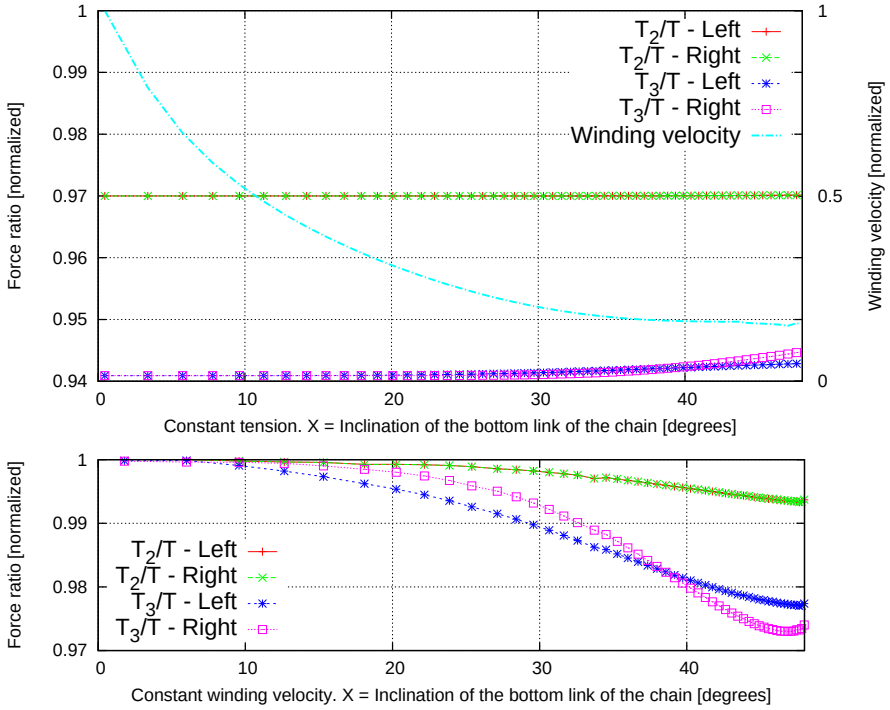


FIGURE 7. Tension propagation.

consider the inclination α_3 of the bottom link of the chain as indication of the posture of the chain.

The diagrams in Fig. 7 plot the tension at each particle level with respect to α_n, which is reported in the X-axes of both diagrams. To make the loss of tension along the chain more evident, the tensions T_2 and T_3 on both sides are normalized to the input value T. **Discussion.** In the first case (1.), a constant tension applied to the chain makes the tension loss along the chain evident. In particular, T_2 on both sides is constantly slightly lower than the actuation tension T. The tension T_3 is also lower than the tension T as expected. However, starting from 30° T_3 (left) and in particular T_3' (right), which is the folding side, slightly increase towards the right most side of the diagram. If we consider the winding velocity – shown normalized to its maximum in the top diagram of Fig. 7 – we notice that as soon as the velocity starts decreasing, the ratio between the tensile forces T_3/T slightly increases. This results from the fact that when the system is reaching an equilibrium point (i.e., velocities almost null), the tensile forces along the tendons also tend to equally redistribute, hence reducing the difference of tension along the chain.

In the second case (2.), a counter-example that confirms the last observation is given in the bottom diagram of Fig. 7. As the chain is actuated at a constant winding velocity, which prevents the system from reaching an equilibrium point, the difference of tensions keeps increasing towards the right side of the diagram. We expect this behavior because the friction between the tendon and the eyelet increases whit the bottom link approaching 45°. In the left side of the diagram, where the bottom link is horizontal and the chain is released, the tensions T_2 and T_3 equal T.

3.2 Loop-Constraint Stability

The dynamic model presented in the previous section is applied to predict the behavior of a chain with an increasing number of particles. We estimate the average distance (drift-off) between the point P and Q (see Fig. 4) that occurs during the execution of the two experiments reported in the previous subsection. We also estimate how an increasing number of particles affects the drift-off when the chain is not actuated – i.e., only subject to gravity.

Table 1 compares different cases. The columns indicate 1. the number of particles in a chain; 2. the type of actuation that can be "static" – i.e., no actuation – or according to the experiments reported in the previous subsection; 3. the total length of the chain measured in its initial posture; 4. the average drift-off computed considering the local drift-off of particles during the

TABLE 1. Drift-Off comparison

Number of Particles	Type of Actuation	Chain Length [mm]	Average Drift-Off [mm]	Normalized Drift-Off [mm/mm]
3	Experiment 1.	115	0.93	0.31
3	Experiment 2.	115	0.94	0.31
3	Static	115	1.0	0.33
35	Static	1334	5.7	0.16
505	Static	17645	39.1	0.07

simulation; 5. the normalized drift-off, namely the total average drift-off normalized to the number of particles.

Discussion The total drift-off increases with the length of the chain, whereas its overall disturbing effect reduces, as the fifth column of Table 1 shows. The table shows that even for a long chain (longer than 17 m) the drift-off at each particle remains within acceptable limits.

An interesting result is that the average drift-off of an actuated chain is slightly lower than the average drift-off measured on the same chain standing still (first three rows of the table). This result is probably due to the fact that when actuated, the distance between P and Q occasionally reduces due to the actuation itself. Consequently, despite the observed deviation, the accuracy of the model applied to an actuated chain is still acceptable.

4 Conclusion

We have briefly introduced the concept of "Shape-Shifting Surface", its structure, and its components. The principal building-block is a piecewise controllable chain that can change its curvature to outline arbitrary 2D curves. In order to enable optimal model-predictive planning and control strategies, necessary to minimize the forces acting on a folding chain and to accurately control the system, a dynamic model is necessary to predict the behaviour of the chain given a specific control input.

The chain is composed of serially connected robotic particles that can fold and hence curve the chain. In order to ensure scalability, fundamental to obtain high resolution of the Shape-Shifting Surface, these particles do not use local actuators (which would imply severe limitation in terms of scalability) and a tendon-driven actuation strategy is instead adopted.

A single pair of tendons actuates the whole chain, namely all the particles in the chain. This makes the kinematic and the dynamic model of the system particularly challenging, especially because particles result to be under-actuated sub-systems.

Kinematic and dynamic models are formulated as well as the friction model that characterizes the interaction between particles and the tendons. The evaluation shows the stability of the loop-constraints formulated in the model, and also analyzes the tension propagation.

REFERENCES

[1] Pillai, P., Campbell, J. D., Kedia, G., Moudgal, S., and Sheth, K., 2006. "A 3d fax machine based on claytronics". In IROS 2006.

[2] Goldstein, S. C., Campbell, J., and Mowry, T. C., 2005. "Programmable matter". *IEEE Computer*.

[3] Ishii, H., Lakatos, D., Bonanni, L., and Labrune, J.-B., 2012. "Radical atoms: beyond tangible bits, toward transformable materials". *Interactions*.

[4] Gilpin, K., Kotay, K., Rus, D., and Vasilescu, I., 2008. "Miche: Modular shape formation by self-disassembly". *Int. Journal of Robotics Research*.

[5] Goldstein, S. C., and Mowry, T. C., 2004. "Claytronics: A scalable basis for future robots". In RoboSphere 2004. 18.

[6] Benbernou, N., Demaine, E. D., Demaine, M. L., and Ovadya, A., 2009. "A universal crease pattern for folding orthogonal shapes". *CoRR*.

[7] Lasagni, M., and Römer, K., 2014. "Force-guiding particle chains for shape-shifting displays". In IROS.

[8] Lasagni, M., and Römer, K., 2016. "Programmable robotic chains: Kinematics and dynamics of a scalable tendon-driven under-actuated multibody system". In AIM 2016.

[9] Featherstone, R., and Orin, D., 2000. "Robot dynamics: equations and algorithms". In ICRA.

[10] Featherstone, R., 1999. "A divide-and-conquer articulated-body algorithm for parallel o(log(n)) calculation of rigid-body dynamics. part 2: Trees, loops, and accuracy". *The International Journal of Robotics Research*.

[11] Shah, S. V., Saha, S. K., and Dutt, J. K., 2013. *Dynamics of Tree-type Robotic Systems*. Springer Netherlands.

[12] Schiehlen, W., 1997. "Multibody system dynamics: Roots and perspectives". *Multibody System Dynamics*.

[13] Flores, P., and Ambrsio, J., 2004. "Revolute joints with clearance in multibody systems". *Computers & Structures*.

[14] Fritzkowski, P., and Kamiński, H., 2010. "Dynamics of a rope modeled as a multi-body system with elastic joints". *Computational Mechanics*.

[15] Lee, J.-J., and Lee, Y.-H., 2003. "Dynamic analysis of tendon driven robotic mechanisms". *Jrnl. of Robotic Systems*.

[16] Zanasi, R., 2010. "The power-oriented graphs technique: System modeling and basic properties". In Vehicle Power and Propulsion Conf.

[17] Nikravesh, P. E., 2007. *Planar Multibody Dynamics: Formulation, Programming and Applications*. CRC Press, Inc.

[18] Baumgarte, J., 1972. "Stabilization of constraints and integrals of motion in dynamical systems". *Computer Methods in Applied Mechanics and Engineering*.

Bibliography

[1] Bioloid kernel description. http://www.robotis.com/xe/bioloid_en.

[2] H. Abelson, D. Allen, D. Coore, C. Hanson, G. Homsy, T. F. Knight, Jr., R. Nagpal, E. Rauch, G. J. Sussman, and R. Weiss. Amorphous computing. *Commun. ACM*, 43(5):74–82, May 2000, DOI: 10.1145/332833.332842.

[3] F. Anooshahpour, I. G. Polushin, and R. V. Patel. Quasi-static modeling of the da vinci instrument. In *Proc. Intelligent Robots and Systems, IROS*, 2014.

[4] M. P. Ashley-Rollman, S. C. Goldstein, P. Lee, T. C. Mowry, and P. Pillai. Meld: A declarative approach to programming ensembles. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems IROS 2007*, 2007, DOI: 10.1109/IROS.2007.4399480.

[5] M. P. Ashley-Rollman, P. Lee, S. C. Goldstein, P. Pillai, and J. D. Campbell. A language for large ensembles of independently executing nodes. In *Proceedings of the International Conference on Logic Programming (ICLP '09)*, July 2009.

[6] J. Bachrach, J. McLurkin, and A. Grue. Protoswarm: A language for programming multi-robot systems using the amorphous medium abstraction. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, 2008.

[7] P. Ball. Make your own world with programmable matter. http://spectrum.ieee.org/robotics/robotics-hardware/make-your-own-world-with-programmable-matter, 2014.

[8] G. Bastos, R. Seifried, and O. Brüls. Inverse dynamics of serial and parallel underactuated multibody systems using a dae optimal control approach. *Multibody System Dynamics*, 2013.

[9] J. Bath and A. J. Turberfield. Dna nanomachines. *Nat Nano*, 2(5):275–284, May 2007.

[10] N. Benbernou, E. D. Demaine, M. L. Demaine, and A. Ovadya. A universal crease pattern for folding orthogonal shapes. *CoRR*, abs/0909.5388, 2009.

[11] E. Benson, A. Mohammed, J. Gardell, S. Masich, E. Czeizler, P. Orponen, and B. Hogberg. Dna rendering of polyhedral meshes at the nanoscale. *Nature*, 523(7561):441–444, July 2015.

[12] P. Bhat, J. Kuffner, S. Goldstein, and S. Srinivasa. Hierarchical motion planning for self-reconfigurable modular robots. In *Proc. IEEE/RSJ Int Intelligent Robots and Systems Conf*, 2006, DOI: 10.1109/IROS.2006.281742.

[13] J. Bourgeois and S. C. Goldstein. Distributed intelligent mems: Progresses and perspectives. *IEEE Systems Journal*, 9(3):1057–1068, Sept 2015, DOI: 10.1109/JSYST.2013.2281124.

[14] J. Bourgeois, B. Piranda, A. Naz, N. Boillot, H. Mabed, D. Dhoutaut, T. Tucci, and H. Lakhlef. Programmable matter as a cyber-physical conjugation. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct 2016, DOI: 10.1109/SMC.2016.7844687.

[15] D. B. Camarillo, C. F. Milne, C. R. Carlson, M. R. Zinn, and J. K. Salisbury. Mechanics modeling of tendon-driven continuum manipulators. *IEEE Transactions on Robotics*, 24(6):1262–1273, Dec 2008, DOI: 10.1109/TRO.2008.2002311.

[16] J. D. Campbell and P. Pillai. Collective actuation. In *RSS 2006 Workshop on Self-Reconfigurable Modular Robots*, August 2006.

[17] A. Castano, A. Behar, and P. M. Will. The conro modules for reconfigurable robots. *IEEE/ASME Transactions on Mechatronics*, 7(4):403–409, Dec 2002, DOI: 10.1109/TMECH.2002.806233.

[18] H. Chaudhary and S. K. Saha. Minimization of constraint forces in industrial manipulators. In *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007.

[19] C. C. Cheah, S. P. Hou, and J. J. E. Slotine. Region-based shape control for a swarm of robots. *Automatica*, 45(10):2406 – 2411, 2009, DOI: http://dx.doi.org/10.1016/j.automatica.2009.06.026.

[20] H.-L. Chen, D. Doty, D. Holden, C. Thachuk, D. Woods, and C.-T. Yang. Fast algorithmic self-assembly of simple shapes using random agitation. In *International Workshop on DNA-Based Computers*. Springer, 2014.

[21] K. C. Cheung, E. D. Demaine, J. Bachrach, and S. Griffith. Programmable assembly with universally foldable strings (moteins). *IEEE Transactions on Robotics*, 27(4):718–729, 2011.

[22] D. Christensen, E. Østergaard, and H. Lund. Metamodule control for the atron self-reconfigurable robotic system. In *Proceedings of the 8th Conference on Intelligent Autonomous Systems*, 2004.

[23] D. N. Coore. *Botanical Computing: A Developmental Approach to Generating Interconnect Topologies on an Amorphous Computer*. PhD thesis, Massachusetts Institute of Technology, 1999.

[24] M. De Rosa, J. Campbell, P. Pillai, S. Goldstein, P. Lee, and T. Mowry. Distributed watchpoints: Debugging large multi-robot systems. In *Proc. IEEE Int Robotics and Automation Conf*, 2007, DOI: .2007.364049.

[25] M. De Rosa, S. Goldstein, P. Lee, J. Campbell, and P. Pillai. Scalable shape sculpting via hole motion: motion planning in lattice-constrained modular robots. In *Proc. IEEE Int. Conf. Robotics and Automation ICRA 2006*, 2006, DOI: .2006.1641915.

[26] M. De Rosa, S. C. Goldstein, P. Lee, J. D. Campbell, and P. Pillai. Programming modular robots with locally distributed predicates. In *Proceedings of the IEEE International Conference on Robotics and Automation ICRA '08*, 2008.

[27] M. De Rosa, S. C. Goldstein, P. Lee, P. Pillai, and J. D. Campbell. A tale of two planners: Modular robotic planning with ldp. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS '09)*, October 2009.

[28] E. P. DeBenedictis. It's time to redefine moore's law again. *Computer*, 50(2):72–75, Feb 2017, DOI: 10.1109/MC.2017.34.

[29] E. D. Demaine, M. L. Demaine, S. P. Fekete, M. Ishaque, E. Rafalin, R. T. Schweller, and D. L. Souvaine. Staged self-assembly: nanomanufacture of arbitrary shapes with o (1) glues. *Natural Computing*, 7(3):347–370, 2008.

[30] Y. Emek and R. Wattenhofer. Stone age distributed computing. In *Proceedings of the 2013 ACM Symposium on Principles of Distributed Computing*, PODC '13, New York, NY, USA, 2013, DOI: 10.1145/2484239.2484244. ACM.

[31] V. Falkenhahn, T. Mahl, A. Hildebrandt, R. Neumann, and O. Sawodny. Dynamic modeling of constant curvature continuum robots using the euler-lagrange formalism. In *Proc. IROS*, 2014.

[32] R. Featherstone. A divide-and-conquer articulated-body algorithm for parallel o(log(n)) calculation of rigid-body dynamics. part 2: Trees, loops, and accuracy. *The International Journal of Robotics Research*, 18(9):876–892, sep 1999, DOI: 10.1177/02783649922066628.

[33] R. Featherstone and D. Orin. Robot dynamics: equations and algorithms. In *proceedings of Int. Conf. on Robotics and Automation, ICRA '00*, volume 1, 2000, DOI: 10.1109/ROBOT.2000.844153.

[34] S. Fekete, A. W. Richa, K. Römer, and C. Scheideler. Algorithmic Foundations of Programmable Matter (Dagstuhl Seminar 16271). *Dagstuhl Reports*, 6(7):1–14, 2016, DOI: http://dx.doi.org/10.4230/DagRep.6.7.1.

[35] P. Flores and J. Ambrósio. Revolute joints with clearance in multibody systems. *Computers & Structures*, 2004.

[36] S. Follmer, D. Leithinger, A. Olwal, A. Hogge, and H. Ishii. inform: Dynamic physical affordances and constraints through shape and object actuation. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, New York, NY, USA, 2013, DOI: 10.1145/2501988.2502032. ACM.

[37] P. Fritzkowski and H. Kamiński. Dynamics of a rope modeled as a multi-body system with elastic joints. *Computational Mechanics*, 2010.

[38] K. Fujibayashi, R. Hariadi, S. H. Park, E. Winfree, and S. Murata. Toward reliable algorithmic self-assembly of dna tiles: A fixed-width cellular automaton pattern. *Nano Lett*, 2007.

[39] T. Fukuda and S. Nakagawa. Dynamically reconfigurable robotic system. In *Proceedings of IEEE Int. Conf. on Robotics and Automation*, Apr 1988, DOI: 10.1109/ROBOT.1988.12291.

[40] K. Gilpin, K. Kotay, D. Rus, and I. Vasilescu. Miche: Modular shape formation by self-disassembly. *Int. J. of Robotics Research*, 27:345–372, March 2008, DOI: http://dx.doi.org/10.1177/0278364907085557.

[41] K. Gilpin and D. Rus. Modular robot systems. *Robotics Automation Magazine, IEEE*, 17(3):38 –55, sept. 2010, DOI: 10.1109/MRA.2010.937859.

[42] K. Gilpin, E. Torres-Jara, and D. Rus. Controlling closed-chain robots with compliant sma actuators: Algorithms and experiments. In *12th International Symposium on Experimental Robotics*, New Delhi, India, 2010.

[43] S. C. Goldstein, J. Campbell, and T. C. Mowry. Programmable matter. *IEEE Computer*, 38(6):99–101, 2005.

[44] S. C. Goldstein and T. C. Mowry. Claytronics: A scalable basis for future robots. In *RoboSphere 2004*, Moffett Field, CA, November 2004.

[45] A. A. Gorbenko and V. Y. Popov. Programming for modular reconfigurable robots. *Programming and Computer Software*, 38(1):13–23, 2012, DOI: 10.1134/S0361768812010033.

[46] S. Griffith. *Growin Machine*. PhD thesis, Massachussets Institute of Technology, 2004.

[47] H. Hamann, M. Wahby, T. Schmickl, P. Zahadat, D. Hofstadler, K. Stoy, S. Risi, A. Faina, F. Veenstra, S. Kernbach, I. Kuksin, O. Kernbach, P. Ayres, and P. Wojtaszek. Flora robotica - mixed societies of symbiotic robot-plant bio-hybrids. In *2015 IEEE Symposium Series on Computational Intelligence*, Dec 2015, DOI: 10.1109/SSCI.2015.158.

[48] E. Hawkes, B. An, N. M. Benbernou, H. Tanaka, S. Kim, E. D. Demaine, D. Rus, and R. J. Wood. Programmable matter by folding. *Proceedings of the National Academy of Sciences*, 107(28):12441–12445, July 2010, DOI: 10.1073/pnas.0914069107.

[49] M. Hiller and A. Kecskeméthy. Dynamics of multibody systems with minimal coordinates. In M. F. O. Seabra Pereira and J. A. C. Ambrósio, editors, *Computer-Aided Analysis of Rigid and Flexible Mechanical Systems*. Springer Netherlands, 1994, DOI: 10.1007/978-94-011-1166-9˙3.

[50] S. Hirose, K. Ikuta, and Y. Umetani. *Development of a Shape Memory Alloy Actuator: Performance Assessment and Introduction of a New Composing Approach*. Advanced Robotics, 1989.

[51] K. Hosokawa, I. Shimoyama, and H. Miura. Two-dimensional micro-self-assembly using the surface tension of water. In *Proc. 'An Investigation of Micro Structures, Sensors, Actuators, Machines and Systems'. IEEE Workshop The Ninth Annual Int Micro Electro Mechanical Systems, MEMS '96*, 1996, DOI: 10.1109/MEMSYS.1996.493831.

[52] F. Hou and W. M. Shen. On the complexity of optimal reconfiguration planning for modular reconfigurable robots. In *2010 IEEE International Conference on Robotics and Automation*, May 2010, DOI: 10.1109/ROBOT.2010.5509642.

[53] F. Hou and W.-M. Shen. Graph-based optimal reconfiguration planning for self-reconfigurable robots. *Robotics and Autonomous Systems*, 62(7):1047 – 1059, 2014, DOI: http://doi.org/10.1016/j.robot.2013.06.014.

[54] H. Huzita and B. Scimemi. The algebra of paper-folding. In *First International Meeting of Origami Science and Technology*, 1989.

[55] H. Ishii, D. Lakatos, L. Bonanni, and J.-B. Labrune. Radical atoms: beyond tangible bits, toward transformable materials. *interactions*, 19(1):38–51, 2012, DOI: 10.1145/2065327.2065337.

[56] H. Ishii and B. Ullmer. Tangible bits: Towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, New York, NY, USA, 1997, DOI: 10.1145/258549.258715. ACM.

[57] Y. Jansen, P. Dragicevic, P. Isenberg, J. Alexander, A. Karnik, J. Kildal, S. Subramanian, and K. Hornbæk. Opportunities and challenges for data physicalization. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015.

[58] P. Jantapremjit and D. Austin. Design of a modular self-reconfigurable robot. In *proceedings of Australian Conference on Robotics and Automation*, 2001, DOI: 10.1.1.12.5956.

[59] M. Jorgensen, E. Ostergaard, and H. Lund. Modular atron: modules for a self-reconfigurable robot. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, sept.-2 oct. 2004, DOI: 10.1109/IROS.2004.1389702.

[60] M. E. Karagozler, J. D. Campbell, G. K. Fedder, S. C. Goldstein, M. P. Weller, and B. W. Yoon. Electrostatic latching for inter-module adhesion, power transfer, and communication in modular robots. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS '07)*, October 2007.

[61] M. E. Karagozler, S. C. Goldstein, and J. R. Reid. Stress-driven mems assembly + electrostatic forces = 1mm diameter robot. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS '09)*, October 2009.

[62] T. Kato, I. Okumura, H. Kose, K. Takagi, and N. Hata. Extended kinematic mapping of tendon-driven continuum robot for neuroendoscopy. In *Proc. Intelligent Robots and Systems, IROS*, 2014.

[63] B. Kirby, B. Aksak, S. C. Goldstein, J. F. Hoburg, T. C. Mowry, and P. Pillai. A modular robotic system using magnetic force effectors. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS '07)*, October 2007.

[64] B. Kirby, J. D. Campbell, B. Aksak, P. Pillai, J. F. Hoburg, T. C. Mowry, and S. C. Goldstein. Catoms: Moving robots without moving parts. In *AAAI (Robot Exhibition)*, Pittsburgh, PA, July 2005.

[65] A. Knaian, K. C. Cheung, M. B. Lobovsky, A. J. Oines, P. Schmidt-Nielsen, and N. Gershenfeld. The milli-motein: A self-folding chain of programmable matter with a one centimeter module pitch. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012.

[66] J. L. Lagrange. *Mécanique analytique*, volume 1. Mallet-Bachelier, 1853.

[67] R. J. Lang. A computational algorithm for origami design. In *Proceedings of the Twelfth Annual Symposium on Computational Geometry*, SCG '96, New York, NY, USA, 1996, DOI: 10.1145/237218.237249. ACM.

[68] J.-J. Lee and Y.-H. Lee. Dynamic analysis of tendon driven robotic mechanisms. *Journal of Robotic Systems*, 20(5):229–238, 2003, DOI: 10.1002/rob.10082.

[69] J. Lengiewicz, M. Kursa, and P. Hołobut. Modular-robotic structures for scalable collective actuation. *Robotica*, FirstView:1–22, 10 2015, DOI: 10.1017/S026357471500082X.

[70] M. J. Martell and J. A. Schultz. Multiport modeling of force and displacement in elastic transmissions for underactuated hands. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014.

[71] M. A. McEvoy and N. Correll. Materials that couple sensing, actuation, computation, and communication. *Science*, 347(6228), 2015, DOI: 10.1126/science.1261689.

[72] P. Mckerrow. *Introduction to Robotics*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1991.

[73] S. Miyashita, M. Kessler, and M. Lungarella. How morphology affects self-assembly in a stochastic modular robot. In *2008 IEEE International Conference on Robotics and Automation*, May 2008, DOI: 10.1109/ROBOT.2008.4543751.

[74] S. Möbes, G. J. Laurent, C. Clevy, N. L. Fort-Piat, B. Piranda, and J. Bourgeois. Toward a 2d modular and self-reconfigurable robot for conveying microparts. In *2012 Second Workshop on Design, Control and Software Implementation for Distributed MEMS*, April 2012, DOI: 10.1109/dMEMS.2012.20.

[75] S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, and S. Kokaji. M-tran: self-reconfigurable modular robotic system. *Mechatronics, IEEE/ASME Transactions on*, 7(4):431–441, 2002, DOI: 10.1109/TMECH.2002.806220.

[76] R. Nagpal. Programmable self-assembly using biologically-inspired multiagent control. In *Proc. 1st Int'l Joint Conf. on Autonomous Agents and Multiagent Systems: part 1*, 2002.

[77] R. Nagpal, A. Kondacs, and et al. Programming methodology for biologically-inspired self-assembling systems. In *AAAI Spring Symposium on Computational Synthesis.*, 2003.

[78] K. Nakagaki, A. Dementyev, S. Follmer, J. A. Paradiso, and H. Ishii. Chainform: A linear integrated modular hardware system for shape changing interfaces. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, New York, NY, USA, 2016, DOI: 10.1145/2984511.2984587. ACM.

[79] A. Naz, B. Piranda, J. Bourgeois, and S. C. Goldstein. A distributed self-reconfiguration algorithm for cylindrical lattice-based modular robots. In *2016 IEEE 15th International Symposium on Network Computing and Applications (NCA)*, Oct 2016, DOI: 10.1109/NCA.2016.7778628.

[80] P. E. Nikravesh. *Planar Multibody Dynamics: Formulation, Programming and Applications*. CRC Press, Inc., 2007.

[81] C. D. Onal, R. J. Wood, and D. Rus. Towards printable robotics: Origami-inspired planar fabrication of three-dimensional mechanisms. In *ICRA*. IEEE, 2011.

[82] E. H. Østergaard, K. Kassow, R. Beck, and H. H. Lund. Design of the atron lattice-based self-reconfigurable robot. *Autonomous Robots*, 21(2):165–183, 2006, DOI: 10.1007/s10514-006-8546-1.

[83] L. S. Penrose. Self-reproducing machines. *Scientific American*, 200(6):105–114, 1959.

[84] W. H. Peterhouse. *Shape Memory Alloys and their Application to Actuators for Deployable Structures*. PhD thesis, University of Cambridge, 1998.

[85] P. Pillai and J. Campbell. Sensing and reproducing the shapes of 3d objects using claytronics. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, SenSys '06, New York, NY, USA, 2006, DOI: 10.1145/1182807.1182859. ACM.

[86] P. Pillai, J. D. Campbell, G. Kedia, S. Moudgal, and K. Sheth. A 3d fax machine based on claytronics. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '06)*, October 2006.

[87] B. Piranda and J. Bourgeois. A distributed algorithm for reconfiguration of lattice-based modular self-reconfigurable robots. In *24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, 2016, DOI: 10.1109/PDP.2016.40.

[88] B. Piranda, G. J. Laurent, J. Bourgeois, C. Clévy, and N. Le Fort-Piat. A new concept of planar self-reconfigurable modular robot for conveying microparts. *Mechatronics*, 23(7):906–915, Oct. 2013, DOI: 10.1016/j.mechatronics.2013.08.009.

[89] V. L. Popov. *Coulomb's Law of Friction*, chapter 11, . Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, DOI: 10.1007/978-3-642-10803-7˙10.

[90] H. S. Raffle, A. J. Parkes, and H. Ishii. Topobo: a constructive assembly system with kinetic memory. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '04, New York, NY, USA, 2004, DOI: 10.1145/985692.985774. ACM.

[91] M. K. Rasmussen, E. W. Pedersen, M. G. Petersen, and K. Hornbæk. Shape-changing interfaces: a review of the design space and open research questions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, New York, NY, USA, 2012, DOI: 10.1145/2207676.2207781. ACM.

[92] D. Rollinson, Y. Bilgen, H. B. Brown, F. Enner, S. Ford, C. Layton, J. Rembisz, M. Schwerin, A. Willig, P. Velagapudi, and H. Choset. Design and Architecture of a Series Elastic Snake Robot. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, September 14-18, 2014*, 2014.

[93] J. Romanishin, K. Gilpin, and D. Rus. M-blocks: Momentum-driven, magnetic modular robots. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, Nov 2013, DOI: 10.1109/IROS.2013.6696971.

[94] M. Rubenstein, C. Ahler, and R. Nagpal. Kilobot: A low cost scalable robot system for collective behaviors. In *2012 IEEE International Conference on Robotics and Automation*, May 2012, DOI: 10.1109/ICRA.2012.6224638.

[95] M. Rubenstein, A. Cornejo, and R. Nagpal. Programmable self-assembly in a thousand-robot swarm. *Science*, 345(6198):795–799, 2014, DOI: 10.1126/science.1254295.

[96] R. Rubien. Daisy chain communication protocol for chains of robotic particles forming shape-shifting displays. Master's thesis, Graz University of Technology, 2016.

[97] D. Rus and M. Vona. Self-reconfiguration planning with compressible unit modules. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 4, 1999, DOI: 10.1109/ROBOT.1999.773975.

[98] H. Sagan. *Space-filling curves*. Springer Science & Business Media, 2012.

[99] E. Şahin. Swarm robotics: From sources of inspiration to domains of application. In *International Workshop on Swarm Robotics: SAB 2004*. Springer Berlin Heidelberg, 2005, DOI: 10.1007/978-3-540-30552-1˙2.

[100] W. Schiehlen. Multibody system dynamics: Roots and perspectives. *Multibody System Dynamics*, 1997.

[101] K. Schwab. *The fourth industrial revolution*. Penguin UK, 2017.

[102] R. Seifried and W. Blajer. Analysis of servo-constraint problems for underactuated multibody systems. *Mech. Sci.*, 4(1):113–129, feb 2013, DOI: 10.5194/ms-4-113-2013.

[103] S. V. Shah, S. K. Saha, and J. K. Dutt. *Dynamics of Tree-type Robotic Systems*. Springer Netherlands, 2013.

[104] K. Stoy and R. Nagpal. Self-reconfiguration using directed growth. In *In Proc. 7th Int. Symp. on Distributed Autonomous Robotic Systems*, 2004.

[105] J. Suh, S. Homans, and M. Yim. Telecubes: mechanical design of a module for self-reconfigurable robotics. In *Proceedings of IEEE Int. Conf. on Robotics and Automation*, 2002, DOI: 10.1109/ROBOT.2002.1014385.

[106] T. Toffoli and N. Margolus. Programmable matter: Concepts and realization. *Physica D: Nonlinear Phenomena*, 47:263–272, 1991, DOI: 10.1016/0167-2789(91)90296-L.

[107] S. Vassilvitskii, J. Kubica, E. Rieffel, J. Suh, and M. Yim. On the general reconfiguration problem for expanding cube style modular robots. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, volume 1, 2002, DOI: 10.1109/ROBOT.2002.1013456.

[108] M. Weller, M. Gross, and S. Goldstein. Hyperform specification: designing and interacting with self-reconfiguring materials. *Personal and Ubiquitous Computing*, 15:133–149, 2011.

[109] M. P. Weller, E. Y.-L. Do, and M. D. Gross. Posey: instrumenting a poseable hub and strut construction toy. In *Proceedings of the 2nd international conference on Tangible and embedded interaction*, TEI '08, New York, NY, USA, 2008, DOI: 10.1145/1347390.1347402. ACM.

[110] P. White, K. Kopanski, and H. Lipson. Stochastic self-reconfigurable cellular robotics. In *Proceedings of IEEE Int. Conf. on Robotics and Automation*, 2004, DOI: 10.1109/ROBOT.2004.1307499.

[111] P. White, M. Posner, and M. Yim. Strength analysis of miniature folded right angle tetrahedron chain programmable matter. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2010, DOI: 10.1109/ROBOT.2010.5509657.

[112] P. White and M. Yim. Scalable modular self-reconfigurable robots using external actuation. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2007, DOI: 10.1109/IROS.2007.4399606.

[113] P. White, V. Zykov, J. Bongard, and H. Lipson. Three dimensional stochastic reconfiguration of modular robots. In *Robotics: Science and Systems*, Cambridge, 2005.

[114] P. J. White. *Miniaturization Methods for Modular Robotics: External Actuation and Dielectric Elastomer Actuation*. PhD thesis, University of Pennsylvania, 2011.

[115] P. J. White, C. E. Thorne, and M. Yim. Right angle tetrahedron chain externally-actuated testbed (ratchet): A shape changing system. In *Proceedings of IDETC/CIE*, San Diego, CA, USA, 2009.

[116] E. Winfree. *Algorithmic Self-Assembly of DNA*. PhD thesis, California Institute of Technology, 1998.

[117] D. Woods, H.-L. Chen, S. Goodfriend, N. Dabby, E. Winfree, and P. Yin. Active self-assembly of algorithmic shapes and patterns in polylogarithmic time. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, ITCS '13, New York, NY, USA, 2013, DOI: 10.1145/2422436.2422476. ACM.

[118] M. Yim, D. G. Duff, and K. D. Roufas. Polybot: a modular reconfigurable robot. In *IEEE International Conference on Robotics and Automation*, 2000, DOI: 10.1109/ROBOT.2000.844106.

[119] M. Yim, W.-m. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. Chirikjian. Modular self-reconfigurable robot systems [grand challenges of robotics]. *IEEE Robotics & Automation Magazine*, 2007, DOI: 10.1109/MRA.2007.339623.

[120] M. Yim, P. White, M. Park, and J. Sastra. Modular self-reconfigurable robots. In R. A. Meyers, editor, *Encyclopedia of Complexity and Systems Science*. Springer New York, New York, NY, 2009, DOI: 10.1007/978-0-387-30440-3˙334.

[121] S. Yim and M. Sitti. Softcubes: Stretchable and self-assembling three-dimensional soft modular matter. *Int. J. Robotics Research*, 2014, DOI: 10.1177/0278364914527630.

[122] E. Yoshida, S. Kokaji, S. Murata, H. Kurokawa, and K. Tomita. Miniaturized self-reconfigurable system using shape memory alloy. In *Proceedings of IEEE/RSJ Int. Conf on Intelligent Robots and Systems*, 1999, DOI: 10.1109/IROS.1999.811704.

[123] R. Zanasi. Power-oriented graphs for modeling electrical machines. In *8th Mediterranean Electrotechnical Conference, MELECON '96*, volume 3, May 1996, DOI: 10.1109/MELCON.1996.551163.

[124] R. Zanasi. The power-oriented graphs technique: System modeling and basic properties. In *Vehicle Power and Propulsion Conf.*, Sept 2010, DOI: 10.1109/VPPC.2010.5729018.

[125] V. Zykov, A. Chan, and H. Lipson. Molecubes: An open-source modular robotics kit. *IROS-2007 Self-Reconfigurable Robotics Workshop*, 2007.

[126] V. Zykov, E. Mytilinaios, M. Desnoyer, and H. Lipson. Evolved and designed self-reproducing modular robotics. *IEEE Transactions on Robotics*, 23(2):308–319, 2007, DOI: http://dx.doi.org/10.1109/TRO.2007.894685.