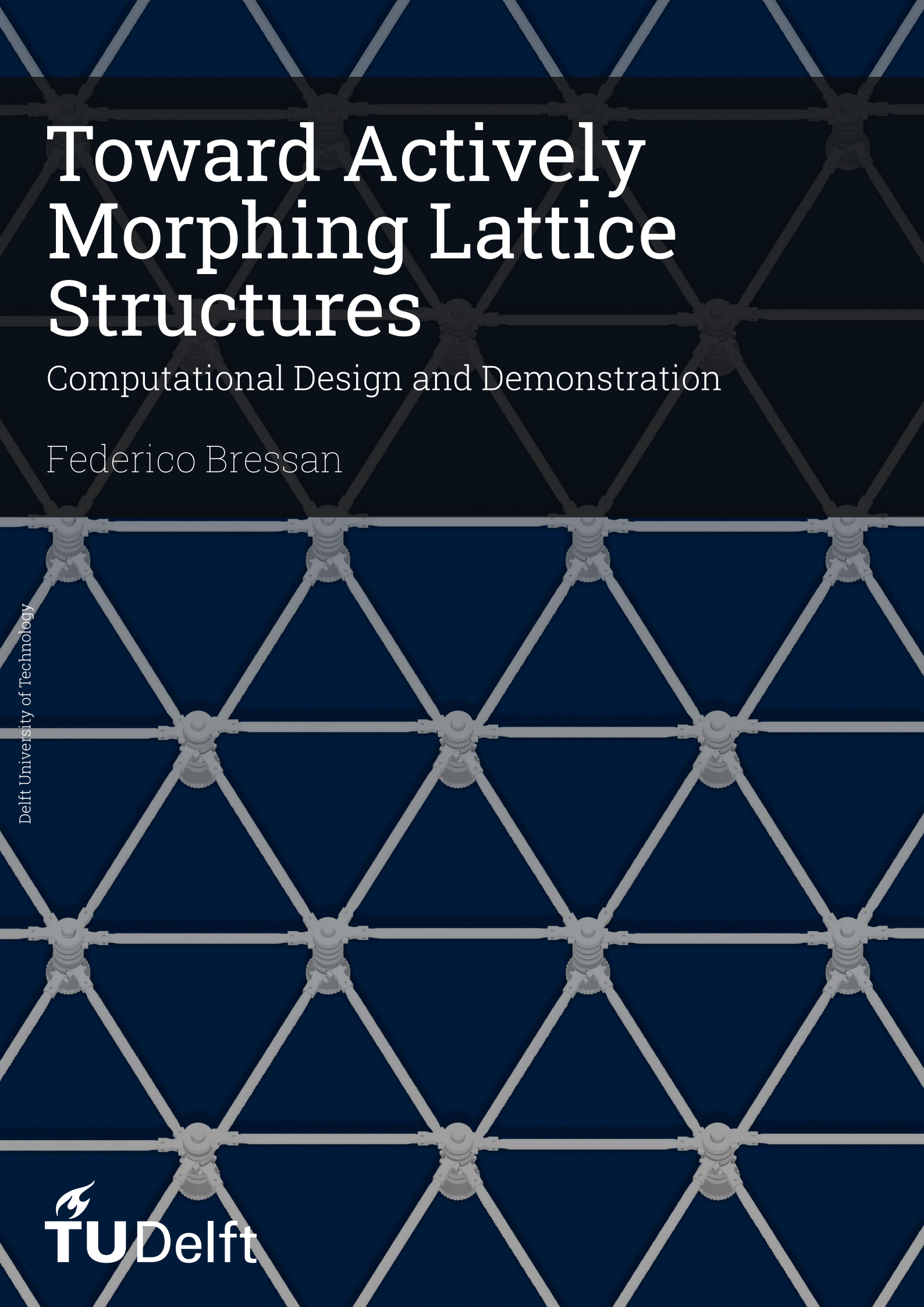
Toward Actively Morphing Lattice Structures

Computational Design and Demonstration

Federico Bressan



Delft University of Technology

TU Delft

Toward Actively Morphing Lattice Structures

Computational Design and Demonstration

by

Federico Bressan

To obtain the degree of Master of Science at the Delft University of Technology, to be defended publicly on Thursday, October 22, 2026 at 15:45.

Student number: 6301088

Project duration: April 13, 2026 - October 22, 2026

Thesis committee: Dr. J. Wu, TU Delft, chair
Dr. M. Özdemir, TU Delft, supervisor

Cover: Rendering of the demonstrator system
Style: TU Delft Report Style, with modifications by Federico Bressan

TU Delft

Abstract

Lattice structures are traditionally designed to be rigid, holding a single shape. Adaptive lattices move between shapes, letting one structure serve changing needs, such as those of a morphing wing rib. Existing actuated lattices follow transformations defined at design, and methods for placing actuators keep all other members at fixed length. None of them releases a member to change length freely, although mixing fixed, actuated and released members could bring a lattice to targets chosen after its design. This thesis pursues that possibility as an inverse design research, for planar pin-jointed lattices of given topology. Given a target curve, it asks which members to fix, actuate or release, how far each actuator must travel, and under what conditions the target can be reached. A computational framework searches the role assignments and computes the actuator travels that bring a set of target nodes onto the curve. Four structures were studied; in the three of these that carry redundant members, no assignment moved the target nodes without a release. These members bind the frame into groups whose lengths cannot change independently, so an actuator stays blocked until releases break every group it belongs to. Which members are released therefore decides whether such a lattice can adapt at all. Tolerance and design preferences, in contrast, change which assignment is selected, not whether one exists. On physical demonstrators built for two of the structures, four computed assignments, with three to seven actuators, ran without adjustment of the computed travels. Taken together, these results state four conditions for adaptation: the roles form a mechanism its actuators control; that motion moves the target nodes toward the curve; an exact fit needs at least one actuator per placement and spacing requirement; and the path avoids folds and stays within travel limits. The first and third can be checked before any search is run. Actuation can thus be computed rather than chosen by hand, a step toward lattices that reconfigure after deployment.

Keywords

adaptive structures; pin-jointed lattices; inverse design; shape morphing; actuator placement; released members

Preface

This project started from an interest in what happens to a structure when it is asked to move. Lattices are usually judged on their mechanical properties – stiffness, strength, energy absorption – and the literature I discovered during the preceding research elective kept arriving at the same conclusion: structures that can deform exist, but each answers to a single required deformation. I wanted to find a way to generalize these transformations and make the design process more systematic, and three elements from my background influenced the choice for this direction. The first was computational design: I wanted to move past parametric modelling as a shaping tool and move towards an application that computes something a designer cannot by hand. The second was kinematic constraints, specifically movable joints, a domain I had touched but never worked at full volume. The third was the capability of translating from digital to physically working systems. Translating a concept into a working product is what I consider a necessary skill for an industrial design engineer, and it is the reason the demonstrator was never optional in this project.

Carrying the project from an idea to completion was not something I could have done alone. I am sincerely grateful to Jun Wu and Mehmet Özdemir for their support and guidance throughout the thesis. Without their input the work would not have reached its current level, and I would not have grown as much as I did. I would also like to thank my family, whose support, even in the most difficult periods, made it possible for me to see this through. Finally, a special thanks goes to my friends and everyone else who stood by me over these two years, whether from afar or close by.

Federico Bressan
Delft, October 2026

Contents

Abstract i

Preface ii

Nomenclature vii

1 Introduction 1
- 1.1 Research Questions . . . 2
- 1.2 Contributions . . . 2
- 1.3 Structure of the report . . . 3

2 Related works 4
- 2.1 Actively reconfigurable structures . . . 4
  - 2.1.1 StarBlocks — Zhao et al. (2023) . . . 5
  - 2.1.2 Modular shape-changing tensegrity-blocks […] – Zhao et al. (2025) . . . 5
  - 2.1.3 Shape morphing hinged truss structures – Sofla et al. (2009) . . . 6
- 2.2 Choosing which members to actuate . . . 6

3 Kinematic preliminaries 7
- 3.1 Maxwell rule and its limitations . . . 7
- 3.2 The compatibility matrix and its rank . . . 8
- 3.3 Self-stress as a release budget . . . 8
- 3.4 Member roles . . . 9
- 3.5 Modelling assumptions . . . 9

4 Demonstrator system 10
- 4.1 Design Constraints . . . 10
- 4.2 Beams . . . 10
- 4.3 Nodes . . . 11
- 4.4 Reference assemblies . . . 11
- 4.5 Control Pipeline . . . 12
- 4.6 Baseline behavior . . . 14

5 Computational framework 15
- 5.1 Architecture . . . 16
- 5.2 Ingestion . . . 17
- 5.3 Target Representation (Curve) . . . 17
- 5.4 The search . . . 18
  - 5.4.1 Enumeration ($L0$) . . . 18
  - 5.4.2 The validity test ($L1$) . . . 18
  - 5.4.3 Usefulness of the available motion ($L1.5$) . . . 19
  - 5.4.4 The symmetry quotient (Mirror) . . . 19
- 5.5 Inner Optimization . . . 20
- 5.6 Scoring . . . 21
- 5.7 Storing and re-scoring . . . 22
- 5.8 Verification . . . 23
  - 5.8.1 The singularity threshold as an input . . . 23

6 Evaluation Protocol 24
- 6.1 Case set . . . 24
- 6.2 Reported measures . . . 25
- 6.3 Evaluation Procedures . . . 25

6.3.1 The release comparison . . . 25
6.3.2 The closeness and preference re-scoring . . . 26
6.3.3 The demonstrator comparison . . . 26
6.3.4 Settings across searches . . . 26

7 Results **28**
7.1 Reduction of the assignment space . . . 28
7.2 Configurations reached . . . 31
7.3 The effect of released members . . . 32
7.4 The effect of design preferences . . . 34
7.5 From computational to physical . . . 37
7.6 Answers to the research questions . . . 41

8 Discussion **42**
8.1 Central finding . . . 42
8.2 What the search reduces, and what it does not . . . 43
8.3 Redundancy as a precondition . . . 44
8.4 Role collapsing: one member for all . . . 44
8.5 Applications . . . 45
8.5.1 Application case: section morphing of a wing rib . . . 45
8.5.2 Application case: posture fitting in a mobile bed frame . . . 46
8.6 Limitations . . . 46

9 Conclusion and future work **48**
9.1 Conclusion . . . 48
9.2 Future development . . . 50
9.3 Contributions . . . 50
9.4 AI Usage . . . 51

References **52**

A Design requirements **54**

B Demonstrator components **55**

C The singularity threshold **56**

List of Figures

1.1 Two frames built once and driven to a second shape: wing rib and bed frame 1

2.1 StarBlocks modules (Zhao et al., 2023). 5
2.2 Tensegrity-block modules (Zhao et al., 2025). 5
2.3 Shape morphing hinged truss structures (Sofla et al., 2009). 6

4.1 Active, fixed and released members. 11
4.2 Exploded and assembled view of a Node 11
4.3 Truss W and Panel P assemblies. 12
4.4 Control Pipeline overview. 13
4.5 Baseline behavior of Truss W. 14

5.1 The problem the framework solves, shown on Panel P 15
5.2 Search algorithm overview. 16
5.3 Ingestion steps from drawn beams to graph. 17
5.4 Target representation: NURBS curve to polyline 17
5.5 Two assignments with different outcomes in the validity test 19

6.1 Case set: Truss W (top left), Panel P (top right), Cross L (bottom left), Short C (bottom right). 24

List of Tables

4.1 Compatibility matrix and rank of the two reference assemblies 12
4.2 Calibration values (mm/s) of the linear actuators 13

5.1 Reduction of the search space by generating directly the admissible group. 18

6.1 Case set overview: numbers of nodes and beams, self-stress states, target nodes, and the purpose of each case. 25
6.2 Search settings per case. 27

7.1 Candidates present at each stage of the funnel, per case. 28
7.2 Actuator placement among the best-placing candidates of each case: value and band per live member, with the heatmap colouring each member by its band. 29
7.3 Placement and spacing accuracy of the Rank-1 candidate of each case (weights 1/0/0/0/0, τ = 5 mm, q = 2). Values in millimetres. 31
7.4 Rank-1 assignment of each case before (left) and after (right) actuation. 31
7.5 Best assignment per structure without a release budget and with the release ceiling at s_0. Values in millimetres. 32
7.6 Cross L: Rank-1 assignment at each release ceiling. Values in millimetres. 33
7.7 Candidates present at each stage of the funnel, per release ceiling, for the three structures carrying self-stress. 33
7.8 Sweep A: Rank-1 assignment across the tolerance τ and the aggregation exponent q, with the weights at their default. Values in millimetres. 34
7.9 Sweep B: Rank-1 assignment across weight vectors, with τ = 5 and q = 2 held fixed. Values in millimetres. 35
7.10 Rank-1 configurations of each case under two weight vectors (p/s/c/e/pa). 36
7.11 Truss W, Curve 1: computed (left) and physical (right) configuration before (top) and after (middle) actuation, and the physical result overlaid on the computed one (bottom). Values in millimetres. 37
7.12 Truss W, Curve 2: computed (left) and physical (right) configuration before (top) and after (middle) actuation, and the physical result overlaid on the computed one (bottom). Values in millimetres. 38
7.13 Panel P, Curve 1: computed (left) and physical (right) configuration before (top) and after (middle) actuation, and the physical result overlaid on the computed one (bottom). Values in millimitres 39
7.14 Panel P, Curve 2: computed (left) and physical (right) configuration before (top) and after (middle) actuation, and the physical result overlaid on the computed one (bottom). Values in millimetres. 40
7.15 Evidence and answer for each research question. 41

8.1 Wing rib: Rank-1 assignment for the NACA 0012 to NACA 4412 morph, with the rib before (top) and after (bottom) actuation. Values in millimetres. 45
8.2 Bed frame: Rank-1 assignment for the reclined contour, with the frame before (left) and after (right) actuation. Values in millimetres. 46

Nomenclature

Abbreviations

DOF	Degree of freedom
JSON	JavaScript Object Notation
NURBS	Non-uniform rational B-spline
PETG	Polyethylene terephthalate glycol
PLA	Polylactic acid
RQ	Research question
SMA	Shape-memory alloy
TPU	Thermoplastic polyurethane

Symbols

Symbol	Quantity	Unit
b, j	Drawn member count (inert members included) and joint count in Maxwell's rule; $j \equiv N$	–
M	Maxwell index	–
N, G, $\lvert G\rvert$	Node count, grounded node set, number of grounded nodes	–
$\boldsymbol{n_{col}}$	Free coordinate count, $2\,(N - \lvert G\rvert)$	–
n, r	Live (non-grounded) member count, rank of the compatibility matrix	–
$\boldsymbol{m_0}$, $\boldsymbol{s_0}$	Independent mechanisms, independent states of self-stress	–
C	Compatibility matrix	–
$\boldsymbol{\sigma_i}$	Singular values of C, $\sigma_1 \geq \sigma_2 \geq \dots$	–
$\boldsymbol{C_f}$, $\boldsymbol{C_{f,a}}$	Compatibility matrix of the fixed members alone; of the fixed and active members	–
f, a, p, g	Member roles: fixed, active, released (passive), grounded	–
u	Unit vector along a member, $(p_j - p_i) \,/\, \lVert p_j - p_i\rVert$	–
$\boldsymbol{n_a}$, $\boldsymbol{n_p}$	Actuator count, release count in an assignment	–
$\boldsymbol{k_{\min}}$, $\boldsymbol{k_{\max}}$	Actuator bracket of the search	–
$\lvert\boldsymbol{L0}\rvert$	Size of the admissible set	–
$\boldsymbol{n_t}$	Number of target nodes	–
P, **StartPct**	Actuation percentage, start percentage	%
$\boldsymbol{P}$	Vector of the actuation percentages $(P_1 \ \dots \ P_{n_a})$	%
stroke	Actuator stroke length	mm
$\boldsymbol{L_a}$	Commanded actuator length	mm
$\boldsymbol{\Delta L_a}$, $\boldsymbol{A_a}$	Commanded length change of actuator a; travel available in the commanded direction	mm
$\boldsymbol{x}\,(\boldsymbol{P})$	Free coordinate vector at the converged configuration	mm
$\boldsymbol{R}\,(\boldsymbol{P})$	Residual vector	mm
$\boldsymbol{d_t}$	Placement error of target node t	mm
$\boldsymbol{\rho_i}$	Spacing residual between consecutive targets	mm
$\boldsymbol{\varepsilon_a}$	Effort residual of actuator a	mm
$\boldsymbol{\lambda_{gap}}$, $\boldsymbol{\lambda_{eff}}$	Residual weights for spacing and effort	–
J, $\boldsymbol{e_a}$	Jacobian of the length constraints at the converged configuration; unit basis vector of actuator a	–
q	Power-mean exponent	–
$\overline{\boldsymbol{e}}_{\boldsymbol{q}}$, $\overline{\boldsymbol{\rho}}_{\boldsymbol{q}}$	Power-mean aggregate of the placement errors; of the spacing residuals	mm
τ	Placement tolerance	mm
$D(e)$	Deadband: 1 for $e \leq \tau$, $\exp\left(-(e-\tau)/\tau\right)$ for $e > \tau$	–
$\boldsymbol{S_{place}}$, $\boldsymbol{S_{spread}}$, $\boldsymbol{S_{count}}$, $\boldsymbol{S_{effort}}$, $\boldsymbol{S_{path}}$	Subscores	–
$\boldsymbol{n_{\min}}$	$\max\left(1,\ n_t\right)$, lower reference of S_{count}	–

Symbol	Quantity	Unit
$w_{place}, w_{spread}, w_{count}, w_{effort}, w_{path}$	Score weights of the five subscores, normalised to sum to one	–
κ_{worst}	Worst ratio of smallest to largest singular value of J along the swept path	–
SING_THRESH	Singularity threshold on κ_{worst}; default 10^{-4}	–
N_{dec}	Decades above SING_THRESH at which S_{path} saturates; $N_{dec} = 2$	–
$p_{\min}$	Release floor	–
MODE_ZERO	Numerical floor for null-space entries	–
ILL_COND	Conditioning floor for the sensitivity factorisation	–

Search stages

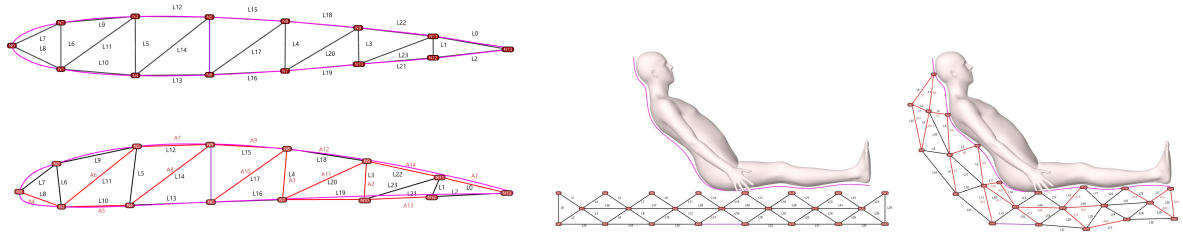
***L*0**	Enumeration of the admissible set from the actuator and release bounds
***L*1**	Validity test (the two-test on the rank of the compatibility matrix)
***L*1.5**	Usefulness of the available motion at the target nodes
Mirror	Symmetry quotient
***L*2**	Inner optimisation of the actuator travels
***L*3**	Verification sweep along the full motion path

1

Introduction

Lattice structures are designed for one geometry, and a new geometry usually requires a new structure. Research has begun to make them active, with actuated pin-jointed trusses (Sofla et al., 2009) and self-connecting deformable blocks (Zhao et al., 2023) that change shape on command, but each of these systems moves through transformations fixed when it is designed. The end state this direction works toward is a structure that reconfigures after deployment, adapting to conditions and requirements that change (Miura and Furuya, 1988).

Two examples, used as illustrations rather than developed designs, show what such adaptation asks of a lattice (Figure 1.1). A wing rib built as a lattice around a symmetric NACA 0012 profile could take the shape of a NACA 4412 to give the wing camber. A care bed whose lattice frame starts flat could recline into a contour shaped to its occupant for long-term care. In both, a frame is built once and asked to take a new shape, and whether it can depends on which of its members are driven, which are freed, and how far each moves. Stated generally, this is an inverse design problem: given a lattice and a target geometry, find the members to actuate, the members to release and the travel of each actuator.



(a) Wing rib, NACA 0012 to NACA 4412

(b) Bed frame, flat to reclined contour

Figure 1.1: Two frames built once and driven to a second shape. (a) A lattice wing rib drawn on a NACA 0012 profile and driven toward a NACA 4412 profile. (b) A planar bed frame, flat as drawn, driven toward the reclined contour of an occupant. Actuated members in red, fixed members in black, grounded members in purple.

Methods exist that choose where to place actuators, both in load-bearing adaptive structures (Senatore et al., 2019) and in structures that change shape substantially (Sachse et al., 2021). In both, a member either keeps its length or is replaced by an actuator. Releasing a member, so that it neither holds nor drives its length but lets it change passively, is not one of the options.

The option matters in frames that carry more members than they need to stay rigid. In such frames, groups of members have lengths bound to one another: lengthening one member of a group would force the others to change length too, so an actuator placed in a group cannot move while the rest of the group is held. These groups are defined as the frame's states of self-stress, and their number is its degree of static indeterminacy. Releasing a member of the group removes the constraint and lets the

actuator drive the frame, so which members are released can decide whether the actuators move the lattice at all, not only how well it reaches the target.

This thesis treats release as a third role. It develops a computational framework that gives every member of a given planar pin-jointed lattice one of three roles, fixed, actuated or released, and computes the actuator travels that bring a chosen set of target nodes onto a target curve; a complete set of roles is called an assignment. A physical demonstrator executes the result. The roles are set when the structure is assembled, after the computation, and only the actuator travels are issued afterwards; a structure that switches roles in the field after deployment is the later step this work prepares.

1.1. Research Questions

A pin-jointed frame may be able to move without any of its members changing length; such a motion is called a mechanism. Pellegrino and Calladine (1986) showed how to count a frame's mechanisms from the rank of its compatibility matrix, the method that relates node movements to changes in member length. This thesis builds on their formulation. A mechanism, however, is necessary but not sufficient. A structure that moves may not move the nodes that matter, or may not move them toward the target; it may reach the target only through a configuration that cannot be built, or in several ways with no basis for choosing between them. What remains undefined is when mobility becomes useful, targeted and buildable in a lattice supplied by a user, and what releasing members changes in that answer.

The aim is to characterize those conditions by examining three decisions taken within a given topology. The decisions are where actuation is placed, whether released members are used and how many the structure can afford, and how acceptable closeness to the target and the remaining design preferences are stated. Additionally, a test conducted is whether the computed result can be run on hardware. An assignment is called a driven mechanism when the frame moves in exactly as many independent ways as it has actuators and stays rigid when they are held. The overarching research question is:

Under what conditions can a pin-jointed lattice of given topology be driven, controllably, toward a prescribed target geometry?

Controllably is taken here to mean that the structure reaches the target without binding, stalling or passing through a configuration where its actuators lose control of its motion. Four sub-questions follow from it:

- **RQ1**: *Within a given topology, which member-role assignments constitute a driven mechanism capable of moving the target nodes, and how does the placement of actuation determine the motion available?*
- **RQ2**: *How does the integration of released members, bounded by the structure's degree of static indeterminacy, change the actuation required to reach a target?*
- **RQ3**: *How do the definition of acceptable closeness to the target and the stated design preferences change which assignment is selected and which configuration is reached?*
- **RQ4**: *Can the assignments and travels prescribed by the framework be executed on a physical demonstrator without adjustment?*

1.2. Contributions

The work makes four contributions:

1. **Conditions for driving a lattice of given topology toward a target**. The work states four conditions an assignment must meet, and shows that in frames with more members than rigidity requires, which members are released decides whether the actuators can move the lattice at all. Both follow from the classical compatibility conditions of the frame; what is contributed is their statement for the inverse problem with three member roles and their test on four planar structures.
2. **A design-automation framework over established kinematic theory**. The rigidity test used throughout is the matrix formulation of Pellegrino and Calladine (1986) with the rank taken by singular value decomposition after the subsequent work of Pellegrino (1993). Neither is new,

what is contributed is a staged search that builds only the admissible role assignments and filters them by rank-based tests, reducing 3^n candidates to a small verified and ranked set.

3. **A separation of measurement from preference**. The search stores a fixed set of measurements for every candidate and the score is computed from that store afterwards, so a new tolerance or design preference re-ranks the candidates without running the search again.
4. **A reconfigurable pin-jointed demonstrator and control pipeline** that execute the framework's output on hardware without intermediate adjustment.

1.3. Structure of the report

Chapter 2 reviews lattice structures that change shape under actuation and the methods available for choosing which members to actuate. Chapter 3 sets out the kinematic preliminaries: Maxwell's rule and its limits, the compatibility matrix and the two indices read from its rank, the member roles and the modelling assumptions. Chapter 4 describes the demonstrator: its members and nodes, the two reference assemblies and the control pipeline that carries the framework's output to the actuators. Chapter 5 presents the computational framework, which takes a drawn lattice and a target curve and returns a verified, ranked set of role assignments with their actuator travels. Chapter 6 defines the evaluation protocol. Chapter 7 reports the results and how they answer to each research question. Chapter 8 discusses what the results show, applies the framework to the wing rib and the bed frame, and states the limitations. Chapter 9 concludes and outlines future developments.

2

Related works

In recent years lattice structures have been at the center of a growing multidisciplinary effort to introduce new properties, design methods, and functional behaviors into cellular architectures. Their rise has coincided with advances in additive manufacturing, since cellular structures reduce the material and energy used and improve the strength-to-weight ratio of the object (Gorguluarslan et al., 2016; Helou and Kara, 2018).

One prominent theme is the application of shape-morphing technologies and reconfigurable metamaterials to cellular structures. By co-designing the latter with smart materials (such as shape-memory alloys, hydrogels) it becomes possible to develop architectures that can be reconfigured between states in response to local triggers or external stimuli. Recent reviews treat programmable morphing and adaptive stiffness as emerging capabilities, while durability, control, and *reversible* morphing remain open challenges (Shen et al., 2025).

In parallel, researchers have begun scaling lattice structures from the meso- to the macro-scale. Several studies (Bironi, 2022; Jenett et al., 2016; Gawde, 2021) focus on adapting established mesoscale cell topologies to increased dimensions, using either additive manufacturing (Bironi, 2022; Kim, 2020) or conventional construction materials and processes (Feng et al., 2016; Jenett, 2020). Although these works differ in application and scope, they collectively demonstrate the feasibility of deploying discrete cellular structures at architectural and engineering scales.

A limitation shared across these efforts is the reliance on application-specific, predetermined geometric planning. Even when shape-morphing materials are integrated, the range of achievable transformations is fixed at the design stage. This points to an opportunity for unit cells whose geometry is not locked to a single target state, and ultimately toward structures that can be reconfigured after deployment to meet new requirements without redesign. Several projects have begun to explore this direction, and their contributions are reviewed in the following section.

2.1. Actively reconfigurable structures

Research on lattice structures that deform geometrically is related to the field of kinematics: large truss structures that can perform transformations thanks to actuators and case-specific solutions already exist. The main limitation (the predetermination of which states and configurations those structures can reach) is the driver for many projects that try to break this constraint. The first two papers tackle the challenge with an approach focused on deformable cellular units that, when assembled in a structure, can perform complex transformations and movements. The last one uses a structure-level vision, with much attention given to kinematic principles and feasibility with larger assemblies.

2.1.1. StarBlocks — Zhao et al. (2023)

StarBlocks, developed by Zhao et al. (2023), presents a system of soft, SMA-based self-connecting blocks that can be assembled into deformable lattice structures. Each StarBlock unit is capable of both structural connection and active deformation (figure 2.1), allowing the assembled lattice to change its global shape without requiring external actuators or connectors. The blocks communicate mechanically through their connection interfaces, and the system has been demonstrated producing a range of global shape changes from flat configurations to curved and three-dimensional forms.

StarBlocks is particularly relevant as a precedent because it addresses both the voxel-level and the shape-level concerns that structure the present thesis. At the voxel level, it demonstrates how a single repeatable unit can embed actuation capability without sacrificing its connectivity function. At the shape level, it shows how the aggregation of locally deformable units can produce globally morphed assemblies. The primary limitation of StarBlocks, from the perspective of structural applications at larger scales, is its reliance on shape memory alloys (SMA) actuation, which may not provide the load-bearing capacity required in architectural or civil engineering contexts.

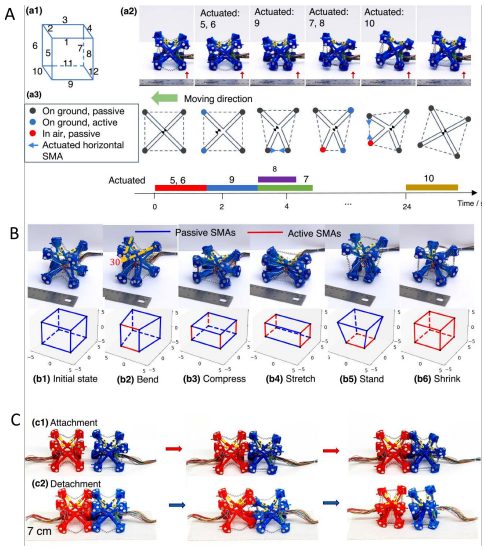


Figure 2.1: StarBlocks modules (Zhao et al., 2023).

2.1.2. Modular shape-changing tensegrity-blocks [...] — Zhao et al. (2025)

In a follow-up to StarBlocks, Zhao et al. (2025) present a system of modular, shape-changing blocks designed to integrate multiple functions within a single unit. Each block is a tensegrity structure consisting of eight rods extending from a flexible TPU central joint, with twelve motors controlling the cables, hence the geometry.

The modules are battery-powered and weigh under 1.2 kg at a side length of approximately 52 cm. Magnetic alignment and a mechanical coupling system allow blocks to attach and detach, and the assembled structures can change shape after assembly through coordinated cable actuation. A single voxel was shown to withstand ∼153 N before failure of the carbon-fibre rods (Zhao et al., 2025).

This work advances the direction established by StarBlocks in two important ways. Firstly, it replaces SMA actuation with cable-driven motor systems, which offer broader control and higher load capacity. Secondly, it scales the concept from bench-top lattice to human-scale infrastructure. The primary limitations, from the perspective of the present research, are two: the magnetic coupling, although it eases connection between units, is weaker than traditional jointing solutions, and the tensegrity topology offers lower stiffness and positional precision compared to rigid pin-jointed truss frameworks.

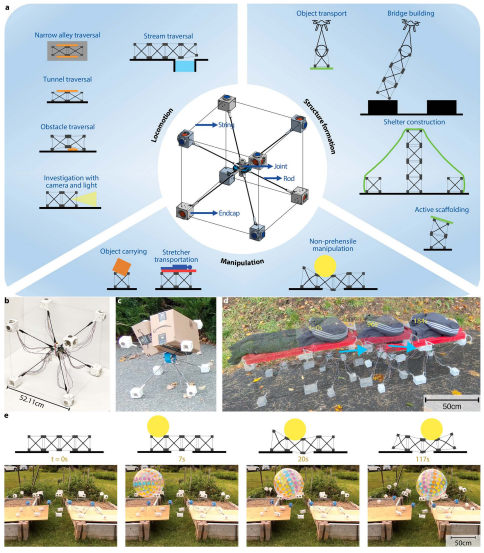


Figure 2.2: Tensegrity-block modules (Zhao et al., 2025).

2.1.3. Shape morphing hinged truss structures – Sofla et al. (2009)

Sofla et al. (2009) propose shape-morphing pin-jointed truss structures built from tetrahedral unit cells connected by a custom spherical joint that permits free rotation between adjacent cells. By replacing as many struts as the mechanism's degrees of freedom with linear actuators, the resulting structure is converted from a mechanism to a statically determinate, morphing truss. The authors demonstrate both linear (beam-like) and planar (plate-like) configurations: a linear hinged truss composed of eight tetrahedra and seven actuators capable of bending and twisting, and a planar hinged truss using a tiling of 36 bi-pyramids.

Maxwell's criterion is used explicitly to size the actuator count to the structure's DOF. An experimental prototype validates the approach using shape-memory alloy wires as linear actuators, achieving rotation angles above 10° per unit cell (Sofla et al., 2009).

This paper operates squarely within the pin-jointed truss framework rather than the soft-body or tensegrity spaces of the projects discussed above. Two of its contributions are particularly important: the use of Maxwell's rule to determine how many members must become actuators to achieve a target number of DOFs, and the choice to add exactly as many actuators as the mechanism has degrees of freedom, so that the actuated truss is statically determinate and has no redundancy that could absorb an actuator's travel. Its principal limitation is that the design is locked to one topology and one actuation strategy, with no generalization approach to arbitrary lattice topologies, selecting which members to actuate.

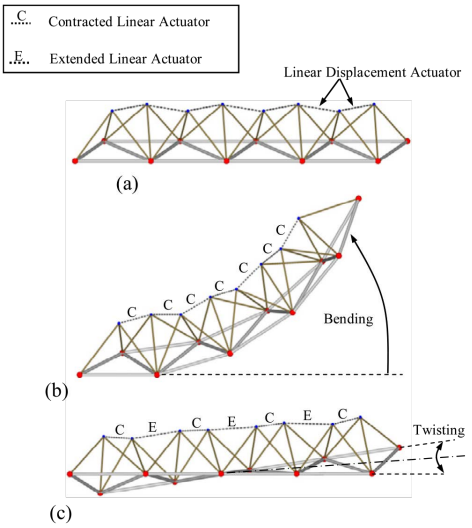


Figure 2.3: Shape morphing hinged truss structures (Sofla et al., 2009).

2.2. Choosing which members to actuate

The question of how to choose which members are to be actuated has been addressed in the field of adaptive structures. Senatore et al. (2019) planned actuator positions together with member sizes so that a load-bearing structure uses as little energy as possible over its life. Instead, Sachse et al. (2021) looks for the smallest set of actuators that can move a truss from one shape to another: they try combinations of actuators, starting from few and adding more, until the target shape can be reached without deforming the remaining members. Their examples show that the number needed depends both on the structure and the target shape.

In both, a member either keeps its length or is replaced by an actuator. A member that is simply freed, carrying no length constraint and not driven, is not considered. This thesis adds this third role, the released member, and asks what it changes in the choice of actuators, which is the subject of RQ2.

3

Kinematic preliminaries

This chapter fixes the definitions that the rest of the thesis uses so that they need not be re-derived later. It states the counting criterion conventionally applied to pin-jointed frames and explains why it is not sufficient for the questions asked in this work, introduces the compatibility matrix and the two indices read from its rank, establishes what those indices imply for the mobility that could be given to a frame, defines the four roles a member can be assigned and the notation used for each in an assignment, and sets out the modelling assumptions that bound the results.

3.1. Maxwell rule and its limitations

The categorization that this thesis started from is one of the earliest criteria used to describe pin-jointed frames: Maxwell's criterion (Maxwell, 1864). The first equation below describes planar geometries and the second three-dimensional ones.

$$M = b - 2j + 3 \qquad \text{(3.1a)} \qquad\qquad M = b - 3j + 6 \qquad \text{(3.1b)}$$

Where M is the Maxwell index, b the number of struts and j the number of joints. In the notation used from Section 3.2 onward $j \equiv N$, while b counts every drawn member, grounded/inert ones included, and therefore differs from the live member count n of the matrix formulation (17 against 16 on Truss W, Section 4.4).

The sign of M provides a counting rule for the static and kinematic determinacy of any assembly of struts and nodes. When $M < 0$ the frame is bar-deficient and behaves, in general, like a mechanism: it possesses at least one internal degree of freedom and can deform without changing the length of any of its members. When $M = 0$ the bar count matches the minimum required for rigidity and the frame is generally statically determinate. When $M > 0$ the frame is over-constrained, statically indeterminate, and can sustain states of self-stress.

In certain geometric configurations a frame satisfying $M \geq 0$ may still admit an internal mechanism, which is why determinacy must ultimately be assessed from the rigidity matrix (Asimow and Roth, 1978) and the criterion is necessary but not sufficient. The standard counterexample is an octahedron having $j = 6$ joints and $b = 12$ bars, so the three-dimensional count gives $M = 0$. A generic octahedron is indeed rigid, but Bricard (1897) found special configurations of the same twelve-bar connectivity that form a mechanism without changing any edge length.

The consequence for this thesis is direct. The question the developed framework must answer for every candidate assignment is not how many members it has, but whether it moves in exactly the ways intended and in no others. That is therefore a question that must be answered with different instruments than with the base count.

3.2. The compatibility matrix and its rank

A single decomposition of a frame's compatibility matrix answers two questions: how many independent ways the structure can move, and how many internal states of prestress it sustains. The rank is computed by singular value decomposition once per candidate assignment and is used only to classify the assignment, never to generate geometry. The formulation is that of Pellegrino and Calladine (1986), with the rank taken by singular value decomposition after Pellegrino (1993).

Each member contributes one row, and each free node contributes two columns, one per coordinate axis. For a member joining nodes i and j, with unit vector

$$\mathbf{u} = \frac{\mathbf{p}_j - \mathbf{p}_i}{\| \mathbf{p}_j - \mathbf{p}_i \|} \tag{3.2}$$

the row carries $-u$ in the columns of node i and $+u$ in those of node j. Columns for grounded nodes do not exist and are therefore skipped. The resulting matrix C has n rows, one per live member, and n_{col} columns, two per free node.

The rank of C is obtained by singular value decomposition, which yields a set of non-negative values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(n,n_{col})} \geq 0$ ordered largest first. Each value measures the extent of the matrix's action along one direction, so a zero value marks a direction in which the members impose no constraint. The rank therefore reads as the number of non-zero singular values, and from it follow the two indices used throughout the thesis:

$$m_0 = n_{col} - r \tag{3.3a}$$

$$s_0 = n - r \tag{3.3b}$$

Here m_0 is the number of independent mechanisms present in the frame as drawn, and s_0 its degree of static indeterminacy, that is the number of independent states of self-stress. A frame carrying no actuators is classified as isostatic when $m_0 = 0$ and $s_0 = 0$, as rigid but statically indeterminate when $m_0 = 0$ and $s_0 > 0$, and as a mechanism when $m_0 > 0$ and $s_0 = 0$.

3.3. Self-stress as a release budget

The count of s_0 is of particular importance to the development of the framework. Releasing a member removes its row from the compatibility matrix of the starting frame, and removing one row lowers the rank by at most one. A frame carrying s_0 states of self-stress can therefore give up at most s_0 rows and remain rigid when its actuators are held; beyond that it remains a mechanism even with every actuator locked.

Which members may be released, and which may then carry an actuator, follows from the states of self-stress themselves. A state of self-stress is a set of member forces that balance at every node without any external load. A frame has s_0 independent ones, collected as the columns of a matrix $\mathbf{T}$ with one row per member, and any combination of them, $\mathbf{Tx}$, is again a state of self-stress. Keeping only the rows of the released members gives $\mathbf{T}_{\{p\}}$, and only those of the active members $\mathbf{T}_{\{a\}}$. For a frame that is rigid as drawn ($m_0 = 0$), an assignment passes the two-test of Section 5.4.2 when

$$\operatorname{rank} \mathbf{T}_{\{p\}} = n_p \tag{3.4a}$$

$$\mathbf{T}_{\{p\}}\mathbf{x} = \mathbf{0} \;\Rightarrow\; \mathbf{T}_{\{a\}}\mathbf{x} = \mathbf{0} \tag{3.4b}$$

Equations 3.4a and 3.4b answer two different questions. Equation 3.4a concerns the frame with its actuators held: removing the released members must take away only redundancy, so that the frame stays rigid. This holds when each released member removes a different state of self-stress, which is what the rank condition expresses, and since the frame has only s_0 states, at most s_0 members can be released. Equation 3.4b concerns the frame with its actuators free. Releasing a member removes only the states of self-stress it takes part in, that is, those in which it carries a non-zero force. If a state not removed by the releases also involves an actuated member, that actuator sits within a group of members whose lengths are bound to one another, so changing its length would require the fixed members of the group to change length too, which they cannot. The actuator is then blocked and produces no motion. Under Equation 3.4a, a release therefore does not make the frame move by itself; it frees the actuators by removing the states of self-stress impeding them. Together, the two conditions are equivalent to the validity test used by the search (Section 5.4.2).

Three consequences follow. First, without releases, Equation 3.4b requires that no state of self-stress involve an actuated member: only members that carry no self-stress can be actuated, and adding further actuators does not change this. Second, when releases are used, an assignment needs at least as many of them as there are independent states of self-stress involving its actuators. The number alone is not sufficient: the releases must also be positioned on members whose removal clears those states. Third, an isostatic frame ($s_0 = 0$) has no state of self-stress, so no release is possible, and every actuator adds one degree of freedom directly. These statements hold within the assumption used in this work, that each actuator adds an independent degree of freedom. Whether a frame needs releases, and how many, therefore depends on which and how many of its members carry self-stress relative to where the actuators are placed.

3.4. Member roles

Every member gets assigned exactly one of four roles, fixed, active, released or grounded, each defined by what it contributes to the compatibility matrix:

- **A fixed member (*f*)** holds its length at the drawn value. It contributes one row to the matrix and is the default state of every member in an inert structure.
- **An active (actuated) member (*a*)** has its length defined within the stroke of an actuator. It contributes one matrix row when the actuator is held (the actuation is stopped and the element retains its length) and is the driving variable when the actuator is free.
- **A released member (*p*)**, for passive; carries no length constraint within the range of its slider. It therefore contributes no row to either matrix. This absence is intentional: removing the row is what lets the actuators govern the frame's motion.
- **A grounded member (*g*)** has both its endpoints pinned. It imposes no constraint on any free coordinate, its row would be identically zero, and it is therefore excluded from the matrix and from all kinematic arithmetic.

A complete assignment is written as one character per live member, taken from *f*, *a* and *p*, in the member ordering established at ingestion. For example, afffffff denotes a nine-member structure with its first member actuated and the rest fixed.

3.5. Modelling assumptions

Although this thesis tries to stay as close as possible to a real-world scenario, some assumptions were made to simplify the development and the mathematics involved.

- Beams do not bend or stretch.
- The system is purely pin-jointed.
- The joints carry no friction.
- The demonstrator is treated as planar and two-dimensional, even though the demonstrator has verticality to it and the actuation creates forces that tend to incline the rotation axis of the node.
- The tolerance and joint looseness of the physical model are not considered, so that the structure resembles an idealized version as closely as possible.
- Actuators are taken to have a starting length equal to their retracted length plus half their stroke, so that both elongation and contraction are available in the examples.
- The topology is fixed and is an input to the work: the framework changes the roles and lengths of existing members and never adds, removes or repositions one.
- Rigidity is assessed to first order, from the rank of the linearised compatibility matrix at the geometry as drawn.
- The analysis is kinematic. No member sizing, stress, buckling, self-weight, external load or actuator force is computed, so no result speaks to whether a reconfigured structure can carry load.

Where an assumption bears on a result, its consequence is stated where that result is reported; those that constrain the results are collected in Section 8.6.

4

Demonstrator system

This chapter explores what was built for the demonstrator: the constraints the design had to satisfy, the resulting members and nodes, the two assemblies used as reference structures for the remainder of the thesis, the pipeline that carries computed values to the actuators, and the baseline behavior of the assembled system before any computed assignment was applied.

4.1. Design Constraints

The requirements for the demonstrator were organized in two levels, so that decisions taken for the lower elements constrain and influence higher requirements. The first level holds the requirements for beams and nodes, while the second those for the assembled structure, each of which depends on the satisfaction of the requirements beneath it.

For the beams, three requirements mainly govern the design. Every member typology must fit a single common node interface: the node-to-beam junction is the critical boundary between designed parts and off-the-shelf components. A passive member must allow contraction and elongation along its axis. An active member must produce actuation that is both controllable and intentional. Fixed members are treated as the baseline from which the other typologies are derived and carry no dedicated requirement.

Concerning the nodes, the governing requirements are a single rotational degree of freedom about the plane normal, freedom to rotate without obstruction at any time, and a single interface geometry accepting every member connecting geometry. A performance requirement completes the set: each node must carry half the weight of every member connected to it – since no member terminates at a single node.

Walking upwards to the assembly level the leading requirements reduce to three: the assembly must perform planar transformations, it must support and actuate its own weight while resting on a flat supporting surface, and it must be easily reconfigurable between member roles.

The full requirement table, with codes and the dependency mapping between levels, is given in Appendix A.

4.2. Beams

Of the three roles, only the active member requires an externally sourced component, so a linear actuator was selected first and the fixed and passive members were then modeled around it, giving a homogeneous length and node interface across the whole system.

The principal decision during the selection of the actuator was whether the model should include a potentiometer. Such component gives positional feedback on the moving end, at the cost of at least one additional wire and complexity to both the driver and the board. A time-based actuator without positional feedback was therefore selected, with the accepted consequence that positional error accumulates

across successive movements.

The selected model has a retracted length of 135 mm, a stroke of 100 mm, a nominal actuation speed of 8 mm/s and a rated force of approximately 70 N. This was a balanced choice among those available, keeping the actuation force high without sacrificing speed or stroke: at 135 mm the structure sits at the larger end of the range, which makes the transformation easier to observe. Fixing the active element also fixed the interface dimensions that the remaining members inherit.

The fixed member length derives from the actuator's retracted length plus half its stroke, following the starting-position assumption. The passive beam is the fixed member with a longitudinal pocket splitting it into two halves that slide within one another, keeping the same interface. One end of the slider carries a dovetail end-stop that can be inserted into the rail but resists at the end of the run, preventing separation. The released member therefore spans lengths from 169 to 286 mm. The resulting slider is considerably weaker than the fixed member, which is accepted: a released member is driven by the forces supplied by the active members rather than carrying load. All parts print without support.

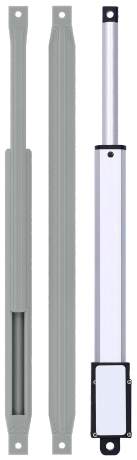


Figure 4.1: Active, fixed and released members.

4.3. Nodes

In this thesis the term *node* refers to a two-part element that represents a pin-jointed connection as closely as a printed assembly allows. To successfully develop this design, obtained through design-prototype-test-improve cycles, three questions had to be answered: what form the node should take, how do members connect to the rotation axis, and how free rotation about that axis is achieved.

The overarching decision that followed was to allow free rotation of every member around the plane normal, and to give each node a single vertical axis. The node in its final form is therefore a two-part assembly of a pin and a set of joints. The pin consists of a base and a staff carrying stepped recesses at fixed heights; the joints clamp onto those recesses. Because the recesses hold a joint at any height independently, a member can be placed on any layer without requiring a joint beneath it to support it. The version used in the demonstrator provides seven layers and therefore accommodates up to seven members per node, though the model is parametric in that number.

The joint is a four-part component: two housings, Top and Bot, enclosing two flaps. Each flap carries an elastic element that lets it open as the joint is slid along the staff and close once the correct height is reached, so that its teeth bite onto the recess and hold the joint in position while leaving it free to rotate. The flaps are printed in PETG, which recovers elastically over repeated cycles where PLA deformed plastically after a few uses. The member interface is made into two separate parts integrated in the housings, and it is coupled with the beams with an M2.5 bolt and nut – a change that resolved the poor geometric precision obtained when the interface was printed as part of the housing.

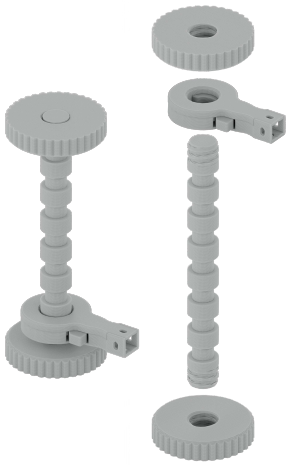


Figure 4.2: Exploded and assembled view of a Node

All components are produced by fused deposition modelling and print without supports, with the flaps in PETG and the remaining parts in PLA. Assembly requires no tool beyond an Allen key for the M2.5 fixings, and the mass of the base holds the node in place while joints are removed.

4.4. Reference assemblies

Two assemblies are used as reference structures for the remainder of the thesis, and each is referred to by a single name in every chapter, table and figure caption: Truss W, a Warren truss, and Panel P.

Truss W was chosen for three reasons. It can be repeated indefinitely along its main axis without becoming over-constrained, and since the structure carries no states of self-stress, it is isostatic: every mobility introduced must be bought with an actuator, as $s_0 = 0$ leaves no release budget at all. It is therefore the structure on which RQ1 is tested in isolation, with RQ2 inactive. The structure is composed of 17 drawn members over 10, of which two nodes are grounded. One member connects those two grounded nodes and is therefore itself grounded and inert (see Section 3.4). Maxwell's rule applied to the drawn structure gives $M = 17 - 2(10) + 3 = 0$, and the matrix formulation applied to the live members gives 16 rows against $n_{col} = 2(10 - 2) = 16$ columns at full rank, so $m_0 = 0$ and $s_0 = 0$. The two results agree, differing only in whether the grounded member should be counted according to the rules.

Panel P was chosen for the opposite property: it carries one state of self-stress, so one member can be released without loss of rigidity. It is therefore the structure on which the passive role, and the physical released member, are first exercised, and on which RQ2 can be analysed. It is an assembly of 16 members over 9 nodes, two of them grounded, with one member inert on the same grounds as above, giving 15 live rows against $n_{col} = 14$ columns at rank 14, so $m_0 = 0$ and $s_0 = 1$. Table 4.1 gives the values obtained from the compatibility matrix and its rank.

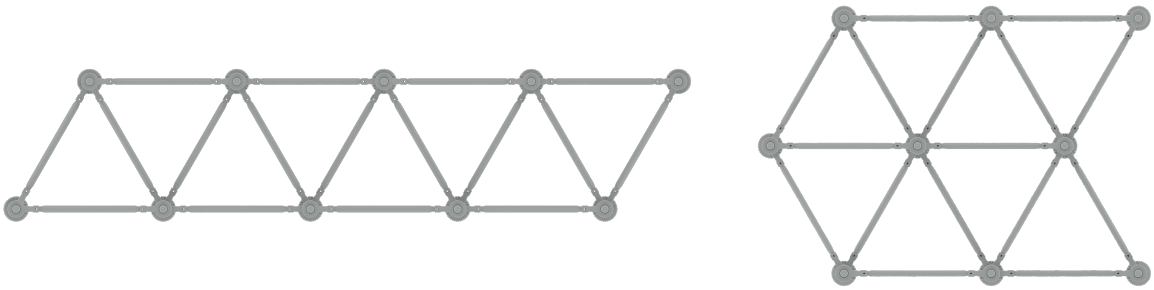


Figure 4.3: Truss W and Panel P assemblies.

Table 4.1: Compatibility matrix and rank of the two reference assemblies

Fixture	Nodes / Beams	n	n_{col}	r	m_0	s_0	M	Verdict
Truss W	10 / 17	16	16	16	0	0	0	isostatic
Panel P	9 / 16	15	14	14	0	1	$+1$	rigid, statically indeterminate

The topologies buildable with this system are limited only by member length, by the topology the case requires, and by the number of members meeting at each node.

4.5. Control Pipeline

The control pipeline sits between the digital and the physical sides of the project, combining the requirements of both and converting the values obtained in the computational environment so that the physical system can act on them.

Although the case shown has been developed for the specific conditions of this thesis, a control pipeline of some kind is required by any adaptive lattice. Whatever stimulus drives the actuation (current, heat, pressure, temperature, or a combination of these) only becomes controlled and finite once a pipeline exists to convert an intended configuration into a signal the actuators can execute.

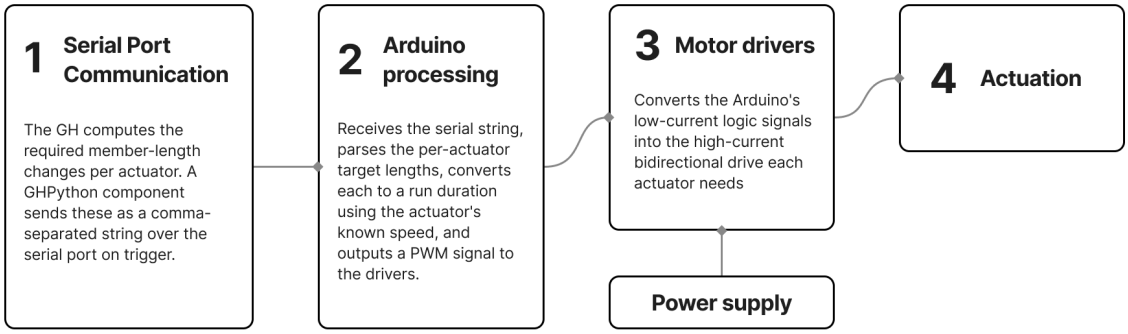


Figure 4.4: Control Pipeline overview.

The pipeline runs in four stages. First, the computational framework calculates the required member-length change per actuator and sends it as a comma-separated string over the serial port. Second, the board parses that string, converts each target length into a time duration using the actuator's calibrated speed, and emits logic-level signals. Third, the motor drivers convert those low-current logic signals into the high-current bidirectional drive each actuator needs. Fourth and finally, the actuators run.

The division of roles within the system follows a deliberately minimal board, acting only as a translator between the digital environment and the drive electronics so that it can be exchanged for another controller without difficulty. The sketch running on it does nothing but parse and dispatch: it splits the received string into its individual elements, checks the sign of each direction value, and matches the values against a pinout table, since positive and negative commands are issued on different pins as the drivers require.

One limitation of Grasshopper influences the design of this stage. The program retains no information across sessions or across changes of value, so if a first movement extended the actuators to 100% and a second required a lower percentage, nothing in the canvas would record where the actuators are. A position record was introduced to overcome this: at every actuation a JSON file is opened (or created if absent) recording positions, stroke length, actuation speed and starting percentage. This makes it possible to issue successive commands without recalibrating the structure each time and to resume the work from the last imput.

The electronics comprise an Arduino UNO R4 WiFi controller and four dual-channel Cytron MDD3A motor drivers, together capable of driving up to eight linear actuators, since each actuator requires two outputs and the analog pins are used as digital outputs alongside the digital ones. The full component list with datasheets is given in Appendix B.

One property of the actuators requires a note, because the pipeline depends on it: the nominal speed does not coincide with the real speed. Every actuator tested returned a speed below the 8 mm/s advertised figure, and the differences between units are large enough that a single actuation speed value cannot serve all of them. Each unit therefore carries its own measured speed in the pipeline.

Table 4.2: Calibration values (mm/s) of the linear actuators

ID	Nominal (mm/s)	Calibrated value	SD	n repeats
1	8	6.82	0.098	5
2		7.07	0.085	
3		7.81	0.138	
4		7.12	0.146	
5		7.63	0.050	
6		6.97	0.078	
7		7.65	0.051	
8		7.18	0.052	

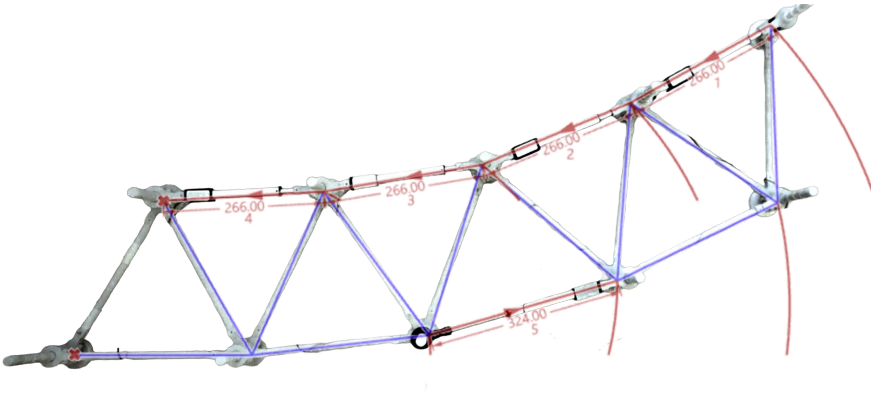
4.6. Baseline behavior

The assembly shown in Figure 4.5 (Truss W) is composed entirely of fixed members and has no internal degrees of freedom, which is the starting configuration the computational framework requires. To verify that movement and adaptation could in fact be enabled in this system, a first run was carried out with linear actuators placed on the top chord and driven from 50% to 70% of their total stroke, and a single actuator on the bottom chord driven from 50% to 80%.



(a) Calibrated position

(b) Actuated position



(c) Actuated position with the computed geometry overlaid

Figure 4.5: Baseline behavior of Truss W.

The photographs, overlaid on the computed configuration, show the same deformation shape: the hardware executes a hand-planned transformation before any computed assignment is applied.

5

Computational framework

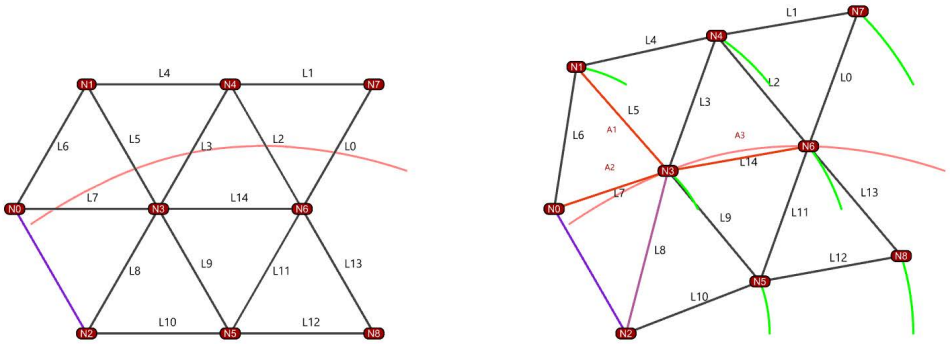


Figure 5.1: The problem the framework solves, shown on Panel P, a 16-member braced panel used as one of the two reference assemblies described in Section 4.4. Left: the lattice as drawn, with its grounded member, the target nodes and the target curve. Right: the same lattice after the framework has given every member a role (three actuated, one released, the rest fixed) and driven the actuators, with the target nodes on the curve. Given the lattice and the curve, the framework decides which members to actuate, which to release, and how far each actuator travels. Actuated members in red, released member in fuchsia, fixed members in black, grounded member in purple.

Figure 5.1 shows the problem this chapter addresses. A lattice is drawn with its grounded nodes, a target curve, and the target nodes that should land on it. The question the frameworks answers is which members are to be actuated, which released, and how far each actuator must travel for the driven lattice to bring its target nodes onto the curve. The wing rib and the bed frame of Figure 1.1 are instances of the same problem.

The forward direction is well understood: given a pin-jointed lattice with a role for every member and a travel for every actuator, the resulting configuration can be computed and assessed. The framework developed here solves the inverse problem, returning the roles and the travels from the lattice and the target.

Two properties make this difficult. The assignment of roles is discrete, so a lattice of n non-grounded members admits 3^n candidates, and most of those are not mechanisms in any usable sense: an assignment must move in as many independent ways as it has actuators when those are free, and become rigid once they are held. That is a condition on the rank of the compatibility matrix (Section 3.2), not on a count of members, so the framework tests it by singular value decomposition rather than by Maxwell's rule, which states of self-stress mislead (Calladine, 1978).

The problem was therefore structured as bilevel. An outer search manages role assignments, and for each valid mechanism it finds, an inner optimisation determines how far to drive the actuators. Survivors of both are verified by simulating the complete motion rather than only its endpoints.

The two levels carry the research questions between them. The outer search (Section 5.4) reduces 3^n assignments to a set that is valid, useful and non-redundant, which is of evidence for RQ1, and the release budget bounding it is where RQ2 acts. The inner optimisation and the scoring (Section 5.5 and Section 5.6) are where acceptable closeness and stated preference enter, and they carry RQ3.

5.1. Architecture

The framework was built as a local server that Grasshopper communicates with through a Hops component, so that the lattice drawn in Rhino can be received by the code as a set of coordinates and member connections, to then be processed outside of such environment, and then returned as index values to use as reference for each assigned role and the values of actuation that the found solution carries.

The code internally is separated into independent layers that are connected in a single one-way dependency, so that no layer depends on the layers above it. At the bottom can be found the elements that analyse the lattice for what it is, regardless of what the program is asked to do: the conversion to a graph representation, the member roles, the compatibility matrix, the forward kinematics and so on. On top of it can be found the toolkit for the target curve, the scoring arithmetic, then the inner optimizer, the outer search and the verification stage. The overarching layer, that is the one that communicates with Rhino, is used only as a conduit for input values and outputs, doing no computing at all. This structure allows to have the lower layers - the scoring for instance - to be free of geometry, and recomputed on request without the need of starting from the top-down again.

The flow of data for the complete search (from Grasshopper back to Grasshopper) is organised as a funnel rather than a loop: candidates pass through costlier filters, each thought to reject the majority of what it reaches: connectivity tests and cheap counting first, secondly a rank test that analyses whether the assignment is a mechanism at all, then a test that understands if the motion provided can be used towards the required direction, ending with the continuous optimization. Finally, only the best candidates go through a full motion sweep to ensure that the first-order mechanism does not reach a singularity or that it does not stiffen (Calladine and Pellegrino, 1991; Kumar and Pellegrino, 2000), as a check that they can be built and driven.

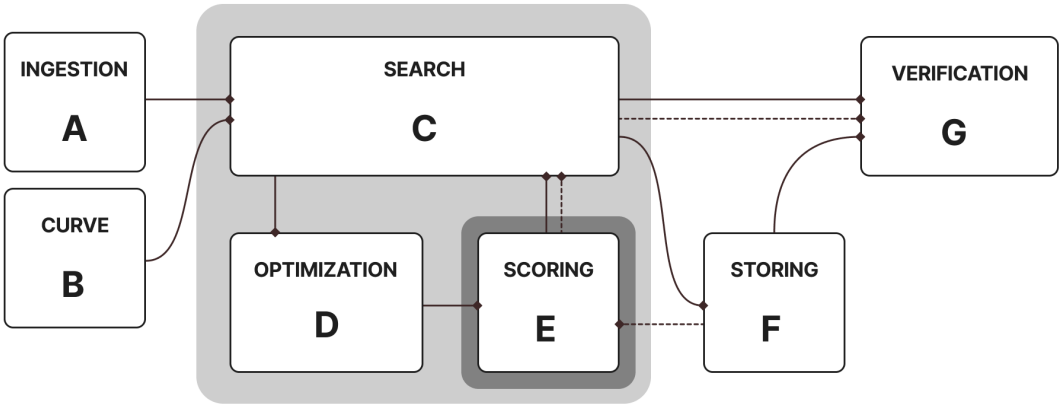


Figure 5.2: Search algorithm overview.

Scoring was made intentionally independent from this pipeline. The search produces a fixed set of measurements per candidate, and preference is applied to those measurements afterwards, so that changing what the designer wants re-ranks stored results instead of re-running the search. This is the basis of Contribution 3 and the reason RQ3 is answerable at all.

5.2. Ingestion

Before any analysis the drawn geometry is converted into a discrete model: numbered nodes and the members connecting them. That numbering is fixed for every later stage and for every table and figure in this thesis, so a member index means the same thing in any stage of the calculations and results.

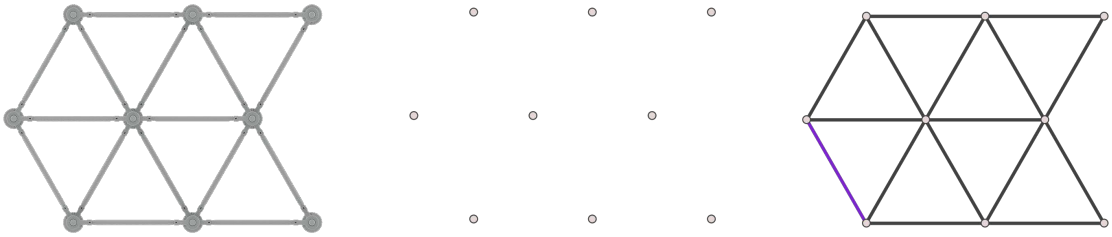


Figure 5.3: Ingestion steps from drawn beams to graph.

Two decisions in this stage matter downstream. Endpoints closer together than a user-set tolerance are treated as one structural node, which is what allows a lattice to be drawn as loose lines rather than as a topologically exact graph. Members with both endpoints grounded are marked inert and excluded from the matrix and from all kinematic accounting, because they constrain no free coordinate; they remain in the output, since they exist physically. This is the distinction between drawn and live member counts used throughout Section 4.4 and Section 3.1. The model handed onward is a set of N nodes of which $|G|$ are grounded, and a member list carrying an inert flag, giving the free coordinate count $n_{col} = 2\,(N - |G|)$ that becomes the column dimension of the compatibility matrix. Inputs from which no structure can be built are refused and the reason inserted in the report.

5.3. Target Representation (Curve)

The search asks only two things of the target curve: how far a target node is from it, and how far apart along it two nodes are. A NURBS curve is computationally heavy, so it is discretised once into an evenly spaced polyline carrying a cumulative arc-length table, and every subsequent distance is read from that table. The default resolution is not relative to the modelling unit, so larger assemblies require it to be raised.

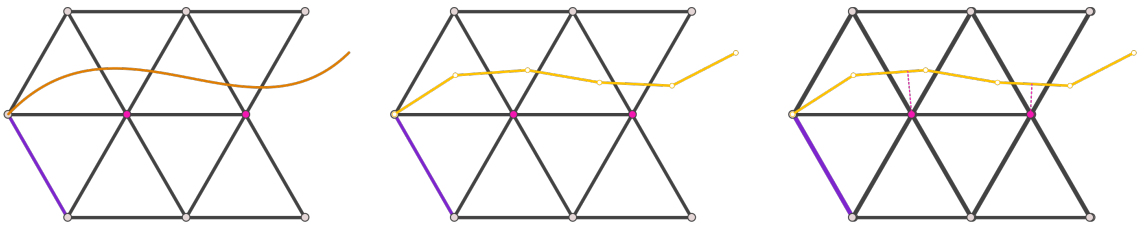


Figure 5.4: The Target representation that converts a NURBS curve to a polyline and measures the distance of the Target Points from it.

Two classes of curve are refused at ingestion. A self-intersecting curve is refused because near a crossing the distance to it is not single-valued, so the closest point has no stable meaning and an ordered chain of contacts along the curve is undefined. A non-monotonic curve is refused because the ordering of its segments would contradict its own parametrization, which would propagate into the spacing residuals of Section 5.5. Both refusals name the location of the offending crossing or break stopping the search before it stars.

5.4. The search

The search is the outer level of the framework: it starts from every way of assigning roles to the members of the lattice and returns the much smaller set of assignments worth optimizing. It is built as a funnel that is walked once, ordered so that the cheapest steps act first and the expensive phases downstream (the optimization, the scoring and the verification), which run once per surviving assignment, see as few candidates as possible. This section covers the four steps of the funnel in the order they are applied: the enumeration of the admissible assignments (L0, Section 5.4.1), the validity test that keeps only driven mechanisms (L1, Section 5.4.2), the check that their motion moves the target nodes toward the curve (L1.5, Section 5.4.3), and the symmetry quotient that removes mirrored duplicates (Mirror, Section 5.4.4).

5.4.1. Enumeration (L0)

Considering a structure with n members, the admissible set of actuators and released members combination (from now on referred to as "assignment") is generated directly by choosing which members will be active and then, from those remaining, which will be passive, so the illegal candidates are never constructed. Its size is known in closed form:

$$|L_0| \;=\; \sum_{n_a=k_{\min}}^{\min(k_{\max},\,n)} \;\sum_{n_p=p_{\min}}^{\min(s_0,\,n-n_a)} \binom{n}{n_a}\binom{n-n_a}{n_p} \tag{5.1}$$

with $k_{\min}$ and $k_{\max}$ the actuator bounds, s_0 the release ceiling and $p_{\min}$ a release floor the user may raise from zero. Knowing this count in advance is itself a design feature: the difference between a search that finishes and one that never does is visible from the counts alone, before any time is spent.

Three conditions influence the set. The actuator count must lie within its bracket; the release count within its own, $p_{\min} \le n_p \le s_0$; and every node must reach the ground through fixed and active members alone, since a released member carries no length constraint and a sub-assembly hanging off passive members would drift during the solve.

The two brackets are not equivalent in what they do, and the difference rules how the framework should be used. Both change which assignments exist, so a search run under one set of bounds cannot be compared with one run under another. But they differ sharply in cost. The inner sum runs over $C\binom{n-n_a}{n_p}$, and its consecutive terms grow as $(n-n_a-n_p+1)/n_p$, so while the release ceiling sits below $(n-n_a)/2$ (which it does in any braced frame, since s_0 is far smaller than the member count) the whole passive range lies on the rising limb of the binomial and the top level dominates the sum.

Table 5.1 shows how the size of the admissible set changes for Truss W and Panel P with an actuator floor $k_{\min}=1$, release floor $p_{\min}=0$ and release ceiling at each structure's own s_0.

Table 5.1: Reduction of the search space by generating directly the admissible group.

$k_{\max}$	Truss W, $\lvert L_0\rvert$	% of 3^{16}	Panel P, $\lvert L_0\rvert$	% of 3^{15}
1	16	0.00004	225	0.0016
2	136	0.0003	1,695	0.0118
3	696	0.0016	7,610	0.0530
4	2,516	0.0058	23,990	0.1672
6	14,892	0.0346	107,073	0.7462
8	39,202	0.0911	216,468	1.5086
10	58,650	0.1362	269,521	1.8783
unbounded	65,535	0.1522	278,512	1.9410
raw 3^n	43,046,721	100	14,348,907	100

5.4.2. The validity test (L1)

An assignment within the context of this thesis is a usable mechanism only if it has exactly as many DOFs as actuators when those actuators are free, and none when they are held. With $C_{\{f\}}$ the com-

patibility matrix of the fixed members alone and $C_{\{f,a\}}$ that of the fixed and active members together- and from both removed the released members' rows:

$$n_{col} - rankC_{\{f\}} = n_a, \qquad n_{col} - rankC_{\{f,a\}} = 0 \tag{5.2}$$

Both halves of the equation are required. A rigid-jointed structure passes the second and fails the first; a pin-jointed chain passes the first at the right count and then collapses under the second. This pair, referenced as the two-test, is the validity criterion of the framework. A passive member carries no length constraint and contributes no row to either matrix, and Equation 3.4 states exactly when removing such rows lets the actuators govern the motion. The first half also fixes the scope of the framework: it requires every actuator to add an independent degree of freedom, and therefore excludes redundant actuation, in which more actuators than degrees of freedom are driven with mutually compatible commands. Such assignments are kinematically possible, but any departure from compatibility is taken up as internal force between the actuators, therefore the exclusion from the framework.

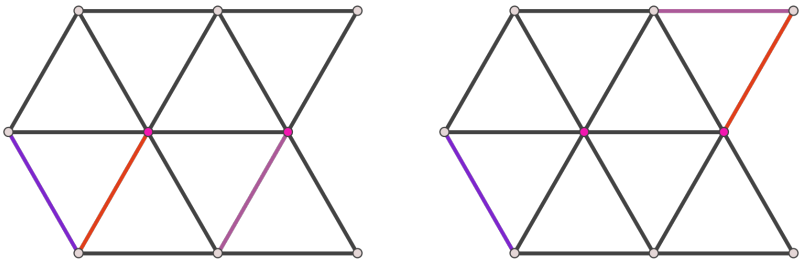


Figure 5.5: Two examples that yield different results in the validity test. On the right an assignment that has $m_0 = 1$ when the actuators are held and therefore fails the test, on the left a valid assignment with $m_0 = 0$ when held.

5.4.3. Usefulness of the available motion (L1.5)

After L1 the remaining enumerated assignments are valid mechanisms that can move, but not necessarily towards the target: an isolated mechanism can live in a far corner of a lattice without disturbing any target node. This stage discards assignments whose available motion leaves the targets stationary and ranks the remainder by how usefully they move them.

Usefulness is directional: motion that slides a target node along the curve does not reduce its distance to the curve; only motion pointing at the curve counts. Each independent motion the mechanism admits is therefore measured against the curve's normal at each target's nearest point, and each candidate carries its best such value into the ranking. This is the cheapest discrimination available before the optimiser runs, and on an isostatic structure it is the only stage that can filter, since the no assignment can be rejected at the two-test.

5.4.4. The symmetry quotient (Mirror)

Following the usefulness of the motion, for each assignment a check for the symmetry quotient is run. Whenever a structure admits symmetries, an actuator on one side and its mirror on the other prescribe the same physical transformation and score identically while remaining two distinct assignments. This cannot be caught by simply caching, because the two symmetric assignments differ character by character. The symmetries are therefore found once, and each orbit of equivalent assignments is represented by a single member.

The qualifying symmetries are those preserving the whole problem, not merely the shape: a map must carry the joint positions onto themselves and preserve the member set, the grounded set and the target set. A mirror that maps the truss onto itself while exchanging a target node for a non-target one does not qualify, because the two assignments it relates do not solve the same problem. The reduction is therefore exact, and every skipped assignment is named in the report offered to the user.

5.5. Inner Optimization

Each surviving assignment is a valid, theoretically useful, mechanism, but how far to drive it is still unknown. This stage answers that for one assignment at a time, and because it runs once per survivor its cost is multiplied by the remaining set count. It is also where acceptable closeness first acquires a meaning, which is why it carries the first half of RQ3.

The design variable is one travel percentage per actuator, P_a, bounded to the physical stroke; the vector of all of them is written $\boldsymbol{P}$. A solution requiring a value outside those bounds is reported as an assignment needing a longer actuator, which is a design finding. Setting the percentages does not directly give node positions: those settle only when every length constraint is simultaneously satisfied, which is computed by driving the mechanism along its path from rest to the commanded state.

The meaning of ”close to a target” is defined by a vector of residuals that the optimizer drives toward zero. Stacking them and minimising the squared norm is what produces a compromise when several requirements cannot be met at once, and the composition of that vector is therefore a design statement rather than a numerical convenience. The vector is composed of three blocks of values:

$$\mathbf{R}\left(\mathbf{P}\right) = \left(\underbrace{d_1, \ldots, d_{n_t}}_{placement}, \underbrace{\sqrt{\lambda_{gap}}\,\rho_1, \ldots, \sqrt{\lambda_{gap}}\,\rho_{n_t-1}}_{spacing}, \underbrace{\sqrt{\lambda_{eff}}\,\varepsilon_1, \ldots, \sqrt{\lambda_{eff}}\,\varepsilon_{n_a}}_{effort} \right) \tag{5.3}$$

- ***Placement*** is the perpendicular distance from each target node to the curve, in millimetres, and is what the search primarily exists to reduce.
- ***Spacing*** exists because landing every node on the curve is not sufficient: a chain of nodes can sit exactly on the curve while grouped at one end of it, so each consecutive pair is also asked to keep its drawn separation measured along the curve. A correction inside this block handles the case of a curve shorter than the chain itself, scaling the requested separations down rather than demanding an unreachable placement, and warning the user that the curve is short.
- ***Effort*** penalises actuator travel; it exists only to break ties in under-determined multi-actuator cases, and its use is reported when it is raised.

Reducing the residuals is done step by step: at each iteration the optimizer changes the actuator percentages P by a small amount, recomputes the configuration and the residuals, and keeps the change if the residuals have decreased. To decide each change, it needs the sensitivity of the configuration, that is, how far and in which direction each target node moves when one actuator percentage changes slightly. There is no closed-form expression for the node positions that could be differentiated directly. What can be differentiated is the condition every converged configuration satisfies: every fixed and active member is at its correct length. Differentiating these length constraints gives the sensitivity as the solution of a single linear system,

$$J\,\frac{\partial \mathbf{x}}{\partial P_a} = +\,2L_a\,\frac{\text{stroke}}{100}\,\mathbf{e}_a \tag{5.4}$$

where J, the Jacobian of the length constraints, records how the length of each fixed and active member changes when each free node moves; it has the same structure as the compatibility matrix of Section 3.2, restricted to those members. J is evaluated once at the converged configuration and is the same for every actuator, so the sensitivities of all actuators are obtained from a single factorization of it. Near a mechanism fold J becomes ill-conditioned: small numerical errors in it produce very large and incorrect sensitivities. The framework detects this and refuses the result rather than passing the optimizer a confidently wrong direction.

With the sensitivity available, each step is computed with the Levenberg–Marquardt method (Levenberg, 1944; Marquardt, 1963). The method treats the residuals as changing linearly with P around the current configuration and moves directly to the percentages that would minimize them under that approximation; this is a Gauss–Newton step. Where the linear approximation is poor, such a step can overshoot, so a damping factor shortens it and turns it toward the direction in which the residuals decrease fastest, a gradient step, which is slower but does not overshoot. The damping is adjusted at

every iteration, lowered after a step that reduces the residuals and raised after one that does not, so the method behaves like Gauss–Newton where the problem is nearly linear and like a gradient descent where it is not. Where the sensitivity is refused, no step can be built from derivatives at all, and the run hands over to a derivative-free simplex method (Nelder and Mead, 1965). That fallback is slower and carries no comparable convergence guarantee, so its use is recorded and reported.

5.6. Scoring

The optimizer hands over the vector of measurements $\mathbf{R}\,(\mathbf{P})$; that the scoring stage turns into a single number so that candidates can be ranked. Five subscores are computed (each between zero and one) and combined by a weighted sum into a score out of a hundred. The scoring is based on a set of three decisions that have been embedded in the framework.

The first is that errors are aggregated before they are downscaled, not after. The placement errors d_t of the n_t target nodes are each combined independently by a power mean whose exponent q is a user input,

$$\overline{e}_q = \left(\frac{1}{n_t} \sum_{t=1}^{n_t} d_t^{\,q} \right)^{1/q}, \quad q \geq 1 \tag{5.5}$$

so that $q = 1$ gives the arithmetic mean, $q = 2$ the root-mean-square, and $q \to \infty$ the largest single error. Forgiving each node first and then averaging would reward a candidate that places most nodes perfectly and one badly, which is an uneven deformation the search should not reward; instead, as q rises the aggregate is dragged toward the worst-placed node, which is how a user can prioritize that every node fall within tolerance rather than just the mean. Values of q below 1 are refused.

The second decision is that errors are downscaled below a tolerance. The aggregated values pass through a deadband

$$D(\overline{e}_q) = \begin{cases} 1 & \overline{e}_q \leq \tau \\ \exp\left(-(\overline{e}_q - \tau)/\tau\right) & \overline{e}_q > \tau \end{cases} \tag{5.6}$$

which is continuous at $e = \tau$ and gives the placement subscore $S_{\text{place}} = D\left(\overline{e}_q\right)$. A node a fraction of a millimetre from the curve, in a search whose stated tolerance is a millimetre, is on the curve for any practical purpose, and penalising it would rank candidates on differences smaller than the structure can be built to. The tolerance τ is therefore a user input expressing what the fabrication can hold, and the scaling is smooth so that candidates on either side of it are not separated by a step. Spacing, whose subscore is named S_{spread}, is treated identically on the gap residuals ρ_i: $S_{\text{spread}} = D\left(\overline{\rho}_q\right)$, with $\overline{\rho}_q$ the same power mean over the $n_t - 1$ consecutive gaps.

The final decision is what the remaining subscores should measure. Actuator countis added to the subscores family: it rewards frugality in the use of actuators, relative to a reference set by the target rather than by the bracket.

$$S_{\text{count}} = clamp\left(\frac{k_{\max} - n_a}{\max\left(1,\, k_{\max} - n_{\min}\right)},\ 0,\ 1 \right), \qquad n_{\min} = \max\left(1, n_t\right) \tag{5.7}$$

Therefore an assignment with n_t actuators, one per target node, scores 1, and one at the ceiling $k_{\max}$ scores 0. The reference n_t is the generic minimum for placement alone: the n_t perpendicular distances are n_t residuals, and an n_a-parameter family of configurations can drive all of them to zero only where $n_a \geq n_t$. The clamp makes the subscore indifferent below that reference, so every assignment with $n_a \leq n_t$ scores 1 and the subscore does not distinguish between them. Where n_t is at or above $k_{\max}$ the denominator falls to one, and every assignment below the ceiling scores 1. The second subscore is Effort, that rewards leaving the travel of a stroke unspent, measured against the authority available in the commanded direction from the start position StartPct:

$$S_{\text{effort}} = \max\left(0,\ 1 - \frac{1}{n_a}\sum_{a}\frac{|\Delta L_a|}{A_a}\right),\quad A_a = \begin{cases} \frac{100-StartPct}{100}\,stroke & \Delta L_a > 0 \\ \frac{StartPct}{100}\,stroke & \Delta L_a < 0 \end{cases} \tag{5.8}$$

As the last subscore, Path Quality rewards a mechanism far from a fold, anchored at the same threshold that decides the singularity verdict in Section 5.8.1:

$$S_{\text{path}} = clamp\left(\frac{\log_{10}\left(\kappa_{\text{worst}}/\text{SING\_THRESH}\right)}{N_{\text{dec}}},\ 0,\ 1\right) \tag{5.9}$$

where κ_{worst} is the worst ratio of smallest to largest singular value of the forward Jacobian J along the swept path and $N_{\text{dec}} = 2$, so that the subscore rises from 0 at the threshold to 1 two decades above it. The five subscores are combined by a weighted sum,

$$\text{Score} = 100\sum_{i=1}^{5} w_i\, S_i \tag{5.10}$$

with $(w_1,\ \ldots,\ w_5) = (w_{\text{place}},\ w_{\text{spread}},\ w_{\text{count}},\ w_{\text{effort}},\ w_{\text{path}})$ being the user weights, normalised to sum to one. The weights are an encoded design preference, so a different weight vector gives a different ranking, which is the object of study in RQ3.

Some of the assignments that get scored, given certain conditions, can saturate four of the five subscores: whenever the placement and the spacing aggregated values are within the tolerance the score will automatically be 1. An actuator count at or below n_t also receives a score of 1, and path quality two decades above the threshold is rewarded with 1 too. Assignments that differ only in quantities that a saturated subscore can no longer discriminate against receive identical scores, and the function cannot define an order among them. Therefore the candidate the search returns first among equal scores is not selected by any criterion of the scoring. Section 8.6 states the consequences for the results reported as Rank-1.

5.7. Storing and re-scoring

The search records a fixed vector of measurements $\mathbf{R}(\mathbf{P})$ for every assignment it solves to a store held alongside the server. Those measurements are the variables the scoring formulas require, and they represent the entire cost in time of computing for that candidate. Separating them from the score is what makes the re-scoring possible.

A search option set by the user belongs to the re-scoring set if it only re-reads stored measurements: the placement tolerance, the aggregation exponent, the five weights, and the singularity threshold. On the other hand, an input belongs to the fingerprint if it changes which candidates exist at all: the geometry, the connectivity, the grounded and target sets, the curve, the stroke, and both brackets for passive elements and actuator. The fingerprint is a hash of every input listed, so that two inputs differing would require a re-run of a complete search. A re-score is permitted only when the stored digest matches the one the current search settings generate. Without that check a stale store could be silently re-ranked and would answer a search the user had stopped requiring.

This binary behavior has two consequences, the first being cost. Changing a design preference re-ranks stored candidates instead of re-running the search, so a user can ask what scoring weights would have selected and get an answer in the time it takes to read the question.

The second consequence is related to validity. The search runs against a time budget, so two queries of nominally the same search could not evaluate identical candidate populations. Any comparison drawn across two runs therefore confounds the effect of a preference change with the computational logic and limitations of the machine. A comparison drawn by re-scoring one frozen population does not, only the variables inside the arithmetic applied to it changes.

5.8. Verification

Everything to this point has judged an assignment on where it starts ($L0$ with the role assignment) and where its solved configuration ends (the Inner Optimization). The remaining check to be satisfied is whether the mechanism can actually be driven between the two. A linkage can begin and end in a buildable configuration while passing through a fold on the way, or while extending a released member further than a physical element could accommodate. This stage drives the mechanism along its whole path in small steps and certifies that it is buildable.

Three things are verified at every step. The forward solve must converge each member; where it cannot the report names the member asked for more length than the geometry can supply. Secondly, the motion must stay clear of a fold, measured by the reciprocal condition number of the forward Jacobian along the path with the worst value over the whole sweep, κ_{worst}, recorded and compared with the singularity threshold SING_THRESH. Finally, each released member must stay within a length range representing what a real released member can do (Section 6.3.4).

An assignment is reported as verified only if all of these hold together and with every actuator inside its stroke length. Only then is it reported as sound and passed forward, with its member indices, its travels and a report carrying the outcome of every test along the pipeline.

5.8.1. The singularity threshold as an input

The threshold at which the sweep declares a fold is a property of the structure being built and of the tolerances it is designed with: how close to a fold is too close depends on the play in the joints, the precision with which the actuators can be commanded, and how much deviation from the computed path the application will tolerate. A value appropriate to a printed demonstrator at the centimetre scale is not appropriate to a machined assembly, and neither is appropriate to a structure an order of magnitude larger.

The threshold (SING_THRESH) is therefore an input supplied per run rather than a constant, and it governs both the reject line here and the floor of the path subscore, so that the verdict a user sees and the score that ranked the candidate cannot disagree. Because it sits in the re-scorable set of Section 5.7, an iteration across threshold values re-ranks a frozen population rather than requiring a fresh search, which is what makes it possible to establish the value empirically at all. The procedure for obtaining the singularity threshold value on a given structure and set of tolerances is expressed in Appendix C.

6

Evaluation Protocol

This chapter defines what it has been measured on the case set, before the presentation of the results. It states the case set and the rationale behind each structure in relation to what is presented to test, defines reported measures, makes the reference values explicit and fixes the procedure by which computed assignments are executed on the demonstrator.

6.1. Case set

Four planar structures form the case set. They were selected to isolate the three decisions the research questions require, so that each can be seen acting separately.

- **Truss W**, the Warren truss described in Section 4.4. Being isostatic with $s_0 = 0$, there is no released members budget, therefore the roles are limited to fixed and active, and every degree of mobility must be bought with an actuator. RQ2 is excluded on purpose, so that actuator placement can be observed alone.
- **Panel P** is the redundant structure with $s_0 = 1$. One release is available, so the search can spend it, and the first form of RQ2 (whether a release is needed before the actuators yield useful motion) can be asked.
- **The crossing-link structure (Cross L)** carries $s_0 = 3$. The release budget is now large enough that the code must choose among releases combinations rather than select one of the available, turning RQ2 from a yes-or-no question into a question of which redundancy to spend.
- **The short-curve case (Short C)** targets a curve shorter than the drawn chain of target nodes. It is the case in which the definition of a reachable target is under strain and belongs to RQ3 for that reason.

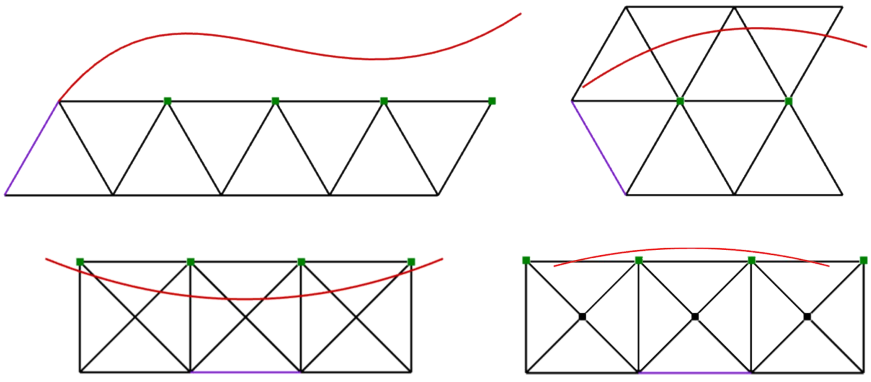


Figure 6.1: Case set: Truss W (top left), Panel P (top right), Cross L (bottom left), Short C (bottom right).

Two cases, Truss W and Panel P, have physical twins and will therefore enter the demonstrator comparison in later sections. Cross L and Short C are evaluated computationally only.

Table 6.1: Case set overview: numbers of nodes and beams, self-stress states, target nodes, and the purpose of each case.

Case	Nodes/Beams	s_0	Target Nodes	Used to test
Truss W	10 / 17	0	4	actuator placement with no release budget (RQ1)
Panel P	9 / 16	1	2	one release available; whether it is required (RQ2)
Cross L	8 / 16	3	4	choice among releases at $s_0 = 3$ (RQ2)
Short C	11 / 20	1	4	short-curve correction; a target under strain (RQ3)

6.2. Reported measures

Every measure reported in Chapter 7 is defined here once, with its unit and with what it does and does not capture.

- **Reduction and placement, RQ1**. The counts surviving each stage of the funnel: the raw space 3^n, the admissible set *L*0, *L*1, *L*1.5, the representatives after the symmetry quotient, the candidates evaluated at *L*2 and those verified at *L*3. Each as an absolute count and as a fraction of 3^n. Alongside them, the placement of actuators within the structure.
- **Achieved configuration measures.** The Rank-1 candidate under placement-only weighting is presented. Inside the tolerance every candidate scores alike on placement, so where the leading candidates fall within τ the one presented is one of several tied candidates rather than the closest placement in the store; outside τ the placement subscore decreases strictly with error and no tie arises. Reported are the perpendicular distance d_t from each target node to the curve, in millimetres, as maximum and mean, and the spacing residuals ρ_i, in millimetres. "Rank-1" denotes the candidate the search returned first, which is not always the highest-scoring candidate in the stored population.
- **Release measures, RQ2**. For each structure with $s_0 > 0$, a series of searches is conducted at otherwise identical settings, one without a release budget and one at each ceiling up to s_0. The comparison is based on which members the search chose to release; and the placement error achieved in each condition.
- **Preference measures, RQ3**. After each search reports the assignment with the closest placement on the curve, a re-scoring is applied with varying weights combinations, and for each is presented: the best-ranking assignment with its actuator and release counts and its effort. Closeness and preference are reported separately, since RQ3 names them as two distinct decisions.
- **Execution measures, RQ4.** For each assignment carried to the demonstrator it was verified whether every actuator completed its prescribed travel, whether any member bound or stalled, whether the released members stayed within their range, and whether the realised configuration corresponds to the computed one on visual overlay. These are recorded as outcomes and the computed placement errors are restated alongside them so that the executed assignment can be identified.

6.3. Evaluation Procedures

This section focuses on detailing the procedures used for evaluation so that later sections do not have to.

6.3.1. The release comparison

Section 3.3 states that an actuator placed on a member carrying self-stress exchanges a fixed length for a commanded one without lowering the rank, so useful mobility appears only once the relevant states of self-stress have been released. For each structure with $s_0 > 0$ a series of searches is run at otherwise identical settings, with the release ceiling set to each value from zero up to the structure's own

s_0. The funnel counts are part of the measurement since the stage at which a candidate set empties distinguishes a structure admitting no mechanism from one admitting mechanisms that do not move the targets. Running a separate search per condition is necessary since the release bounds sit inside the fingerprint of the stored results, so a store built with releases permitted contains candidates the actuator-only condition never generated, and the comparison cannot be drawn from one store.

6.3.2. The closeness and preference re-scoring

RQ3 names closeness and the design preferences as unique variables that influence the output offered by the algorithm, and they must be considered independently. Both are presented by re-scoring a single frozen candidate population rather than by running a search for each condition.

The search runs against a time budget, so two executions of nominally the same search evaluate slightly different populations of candidates – unless the whole enumerated set of genomes is computed. A comparison drawn across two runs might therefore mix the effect under study with the effect of the machine, and neither the tolerance nor the weight vector would be the only thing that had changed between the two columns being compared. Since the tolerance, the aggregation exponent and the five weights all sit in the re-scorable set defined in Section 5.7, one complete search produces a population against which every condition in both sweeps can be evaluated with nothing else varying.

Sweep A, closeness. The resulting population scored across a series of values of the tolerance and of the aggregation exponent, with the five weights held at the search settings throughout. Reported: the best ranking assignment at each value.

Sweep B, preference. The same stored population re-scored across a series of weights combinations. Reported: the same quantities, against the weight vector rather than the tolerance.

Because both sweeps read the same population, the best ranking candidate of one condition is directly comparable with that of another and the crossover between them is attributable to the scoring values that changed.

6.3.3. The demonstrator comparison

A candidate is selected from the ranked output of a completed search. The physical structure is reconfigured to match its assignment, exchanging fixed members for active and passive ones at the member indices the framework returns. The actuators are calibrated so that the starting configuration is known, and the prescribed travels are issued through the pipeline of Section 4.5 in a single command.

What is then assessed is executability. A run is recorded as executed if every actuator completes its prescribed travel without binding or stalling, every passive member accommodates the motion without leaving its sliding range, and the structure arrives at a configuration that corresponds to the computed one – verified with the realised state overlaid digitally on the computed drawing. The Mean d_t and Max d_t values that accompany each physical run are the computed placement errors of the selected candidate, reproduced for reference.

Two assignments per structure are carried through, each the Rank-1 candidate of its own search: one for the target curve used throughout Chapter 7, and an additional one for a second target curve given to the same structure. Applying two prescriptions to one assembly makes it possible to verify that the same lattice, reconfigured at different member indices and driven by different travels, can execute both, so that executability does not rest on a single fortunate assignment. Additionally, the second search for Panel P returns a seven-actuator assignment, which drives the released member under a larger set of active members than the three-actuator case does.

6.3.4. Settings across searches

Two kinds of setting enter a search. The first are held constant across every run reported in this thesis: curve sampling resolution of 1000 steps; verification sweep of 200 steps; null-space floor MODE_ZERO = 10^{-9}; conditioning floor ILL_COND = 10^{-12} scaled by matrix size; passive length bounds of 0.86 and 1.22 times the drawn length; start percentage of 50%, following the assumption of Section 3.5; SING_ThRESH = 10^{-4}; and the scoring inputs at which every search was run, τ = 5 mm, q = 2, the code default weights w = (0.40, 0.20, 0.15, 0.10, 0.15). The re-scoring sweeps vary the tolerance, the

exponent and the weights afterwards on the stored population, so they are search settings here and independent variables there.

The second kind vary per case and are collected in Table 6.2: the actuator bracket, the release floor and ceiling, the stroke, the number of candidates carried into verification and the time budget. Where a case was searched at several release ceilings, the floor and ceiling are listed per run in the order the runs are reported; the ceiling-one run on Cross L also enumerated the actuator-only candidates, none of which passes the two-test, so the difference is without consequence for the results. The time budget is the setting behind the one truncation in the case set: on Short C the admissible set was not exhausted within it, and the counts reported for that structure are those reached when the budget expired.

Table 6.2: Search settings per case.

Case	Search	$k_{\min}$	$k_{\max}$	$p_{\min}$	Ceiling	Stroke	StartPct	τ	q	TopN	Budget (s)
Truss W	Curve 1-2	1	8	–	0	100	50	5	2	3	10,000
Panel P	Curve 1	1	4	0	1	100	50	5	2	3	10,000
	Curve 2	3	8	0	1	100	50	5	2	3	10,000
Cross L	Ceilings 0-3	3	4	0/0/1/1	0/1/2/3	100	50	5	2	3	10,000
Short C	Curve 1	1	7	0	1	100	50	5	2	3	10,000

7

Results

This chapter reports the results that the framework and the demonstrator produced, ordered by research question. Sections 7.1 and 7.2 report the reduction of the assignment space and the configurations reached, and address RQ1. Section 7.3 reports what a release budget buys and addresses RQ2. Section 7.4 reports the two sweeps and addresses RQ3, and Section 7.5 reports the execution of computed assignments on the demonstrators and addresses RQ4. Section 7.6 collects the answers.

7.1. Reduction of the assignment space

For the first half of RQ1, Table 7.1 reports, for each case, the number of candidates present at each stage of the funnel.

Table 7.1: Candidates present at each stage of the funnel, per case.

Case	Raw 3^n	*L*0 (% of 3^n)	*L*1	*L*1.5	Mirror	*L*2	*L*3 (% of 3^n)
Truss W	4.305×10^7	0.09107	39202	39202	–	39202	0.09107
Panel P	1.435×10^7	0.1537	16170	15147	15147	15147	0.1056
Cross L	1.435×10^7	3.123	126590	126320	63217	63217	0.4406
Short C	1.162×10^9	0.103	1.103×10^6	1.103×10^6	–	576528	0.0496

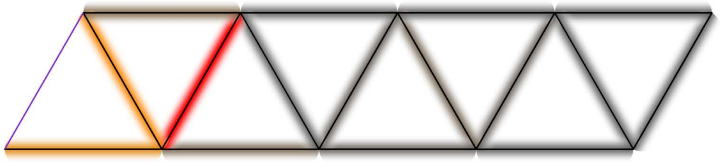
Across the case set the admissible set *L*0 accounts for a fraction of the raw 3^n space, and the candidates reaching the expensive (in computing terms) verification account for an even smaller portion of those. The *L*3 column is an upper bound rather than a count of sweeps: the full verification sweep runs on the TopN = 3 leading candidates of each search (Table 6.2), and the column assumes that every candidate solved at *L*2 would pass it. No swept candidate was rejected, here or at any Cross L release ceiling, so neither the singularity threshold nor the passive length bounds were reached by any candidate that was swept.

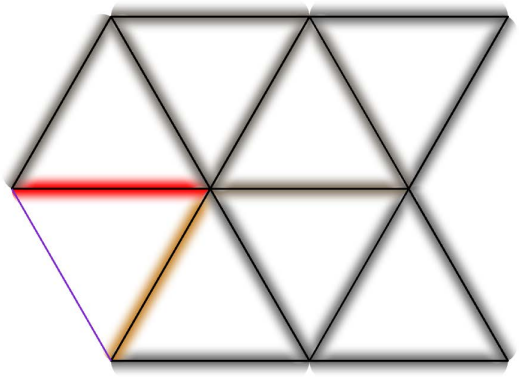
The two gates work differently on the case set structures, and the contrast between Truss W and Panel P – and all the other remaining structures – shows it directly. On Truss W, with $s_0 = 0$, no member can be released, and every admissible assignment is already a valid drivable mechanism, so *L*1 cannot fail, while *L*1.5, the only gate that could reject an assignment, rejected none, and the set reaches the optimiser unfiltered. On Panel P, with $s_0 = 1$, both gates filter: *L*1 rejects 27% of the arriving structures and *L*1.5 a further 6%.

The second half of RQ1 concerns where actuation should be placed, which the counts alone do not answer: every member appears as an actuator in the same number of admissible assignments, so a table of per-member counts over the verified set is constant. Table 7.2 therefore reports placement among the best-placing candidates. Within each actuator-count branch of the set remaining after the funnel, the candidates in the top 5% by placement error are taken, and for each has been calculated n_a/n for a branch of n_a actuators over n live members; the branch values are then averaged. A value

of 1 means placement is indifferent to the member, above 1 that it favours it, below 1 that it avoids it. Table 7.2 lists them per member, and the heatmaps colour each member by its band.

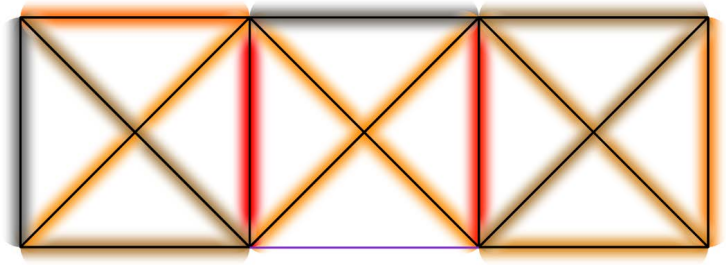
Table 7.2: Actuator placement among the best-placing candidates of each case: value and band per live member, with the heatmap colouring each member by its band.

Case			
Truss W			
	L3	4.4	Strongly favoured
	L6	2.7	Strongly favoured
	L7	1.8	Strongly favoured
	L2	1.0	Neutral
	L5	0.8	Disfavoured
	L9	0.6	Disfavoured
	L0	0.6	Disfavoured
	L12	0.5	Disfavoured
	L8	0.5	Disfavoured
	L1	0.4	Avoided
	L4	0.3	Avoided
	L13	0.3	Avoided
	L14	0.3	Avoided
	L15	0.3	Avoided
	L11	0.3	Avoided
	L10	0.3	Avoided

Case			
Panel P			
	L7	7.8	Strongly favoured
	L8	2.3	Strongly favoured
	L14	1.1	Neutral
	L6	0.6	Disfavoured
	L3	0.6	Disfavoured
	L5	0.6	Disfavoured
	L2	0.6	Disfavoured
	L4	0.6	Disfavoured
	L11	0.2	Avoided
	L0	0.1	Avoided
	L1	0.1	Avoided
	L12	0.1	Avoided
	L13	0.1	Avoided
	L10	0.1	Avoided
	L9	0.1	Avoided

Case

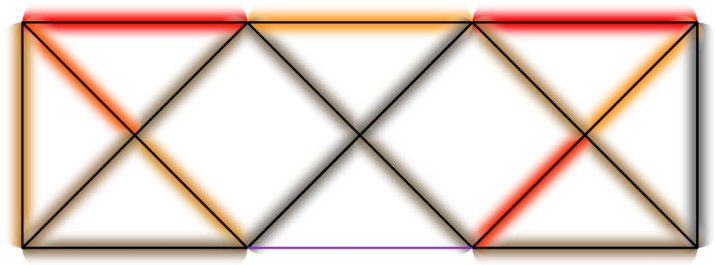
Cross L



L11	2.1	Strongly favoured
L12	1.9	Strongly favoured
L2	1.5	Strongly favoured
L0	1.3	Favoured
L6	1.3	Favoured
L9	1.3	Favoured
L5	1.1	Neutral
L14	0.8	Disfavoured
L8	0.8	Disfavoured
L13	0.6	Disfavoured
L7	0.6	Disfavoured
L10	0.6	Disfavoured
L3	0.5	Disfavoured
L1	0.2	Avoided
L4	0.1	Avoided

Case

Short C



L16	2.7	Strongly favoured
L18	2.5	Strongly favoured
L2	2.3	Strongly favoured
L13	1.9	Strongly favoured
L3	1.5	Strongly favoured
L17	1.3	Favoured
L12	0.9	Neutral
L0	0.9	Neutral
L9	0.7	Disfavoured
L8	0.6	Disfavoured
L6	0.6	Disfavoured
L15	0.5	Disfavoured
L7	0.4	Avoided
L14	0.4	Avoided
L5	0.4	Avoided
L4	0.4	Avoided
L11	0.3	Avoided
L10	0.3	Avoided
L1	0.1	Avoided

On Truss W actuation concentrates at the grounded end: L3 (4.4), L6 (2.7) and L7 (1.8), the two members that tie the first free node to the grounded nodes and the diagonal joining it to the first target, while the values fall with distance from the support. On Panel P the preference is significant: L7 (7.8), the largest value the normalisation allows for the bracket $k_{\min} \dots k_{\max} = 1 \dots 4$, is actuated in every best-placing candidate of every branch, and with L8 (2.3) it forms the pair of members joining the first target to the grounded nodes; while L14, between the two targets, is neutral (1.1). On Cross L the interior verticals L11 (2.1) and L12 (1.9), each joining a grounded node to a central target, lead, followed by the middle-bay diagonals L6 and L9 (1.3), which join the same grounded nodes to the opposite central target. The left–right differences among Cross L's outer members (L0 1.3 against L4 0.1, L2 1.5 against L3 0.5) are properties of which member of each mirror pair the quotient kept. On Short C the preference moves to the top chord: all three segments between the targets are preferred, L16 (2.7), L18 (2.5) and L17 (1.3), together with the outer-bay diagonals L2 (2.3), L13 (1.9) and L3 (1.5).

7.2. Configurations reached

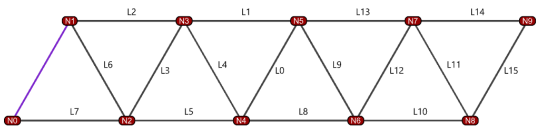
Table 7.3 reports, for the Rank-1 candidate of each case with scoring focused on placement (weights 1/0/0/0/0, τ = 5 mm, q = 2), the placement and spacing accuracy achieved in millimetres. Table 7.4 shows the same four assignments before and after actuation, with the number of actuators each uses.

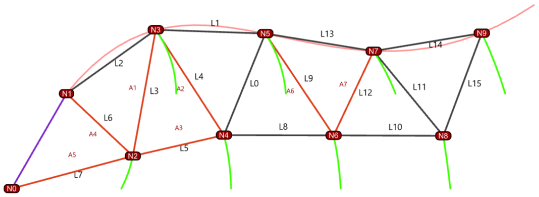
Table 7.3: Placement and spacing accuracy of the Rank-1 candidate of each case (weights 1/0/0/0/0, τ = 5 mm, q = 2). Values in millimetres.

Case	Max d_t	Mean d_t	$\bar{e}_q$	Max gap residual	τ (mm)	Within τ?
Truss W	0.6861	0.5140	0.5252	2.4257	5	Yes
Panel P	Exact placement	∼0	∼0	∼0		Yes
Cross L	14.7674	11.4067	11.8492	6.8867		No
Short C	9.7852	4.6956	5.7927	1.3354		No

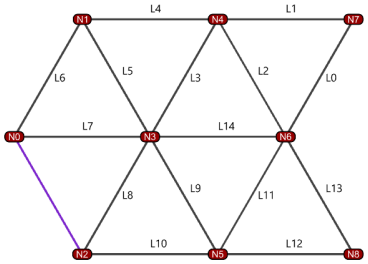
Table 7.4: Rank-1 assignment of each case before (left) and after (right) actuation.

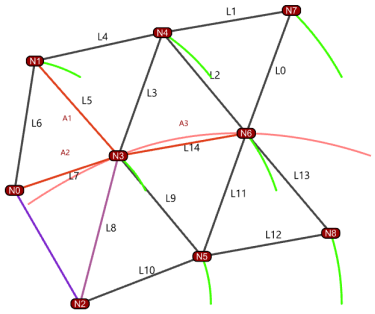
Case	τ	q	Rank-1	n_a
Truss W	5	2	`fffaaaaafaffafff`	7



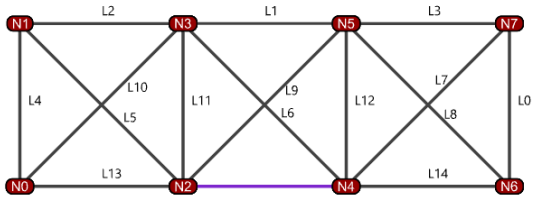


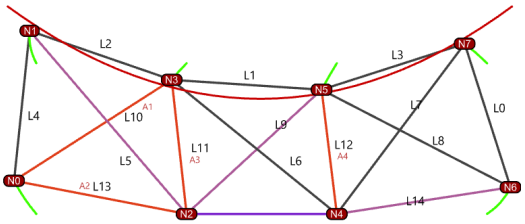
Case	τ	q	Rank-1	n_a
Panel P	5	2	`fffffafapffffa`	3



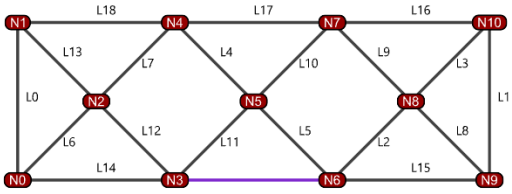


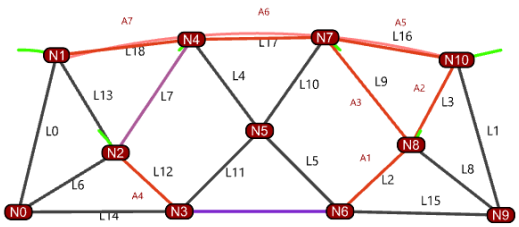
Case	τ	q	Rank-1	n_a
Cross L	5	2	`fffffaffpppaaaf`	4





Case	τ	q	Rank-1	n_a
Short C	5	2	`ffaafffpfaffafffaaa`	7





Two of the four cases placed every target node within the tolerance in force, and two did not. On Truss W the maximum placement error was 0.686 mm against a tolerance of 5 mm, and the aggregate of 0.525 mm sits close to the mean of 0.514 mm, which indicates a set of nodes placed uniformly rather than one node carrying the result. Panel P reached placement below the precision at which the candidate record reports ($<$ 0.01 mm), which the tables mark as exact placement. Both values belong to candidates inside the deadband, where every candidate scores 1 on placement and ties with every other candidate inside it, so neither is necessarily the closest placement in its store: Sweep A returns a Truss W candidate at a mean of 0.24 mm from the same population (τ = 1 mm, q = 2, Table 7.8). Cross L and Short C did not reach the tolerance; there the placement subscore decreases strictly with error, so the score orders their candidates by aggregate error and no tie arises.

Cross L and Short C did not reach the tolerance, with a counting pattern that can be identified. The optimiser of Section 5.5 drives $2n_t - 1$ geometric residuals toward zero, n_t placement distances and $n_t - 1$ spacings, while an assignment with n_a actuators moves the target nodes through an n_a-parameter family of configurations, so every residual can be driven to zero at once only where n_a reaches $2n_t - 1$ and the mechanism's motion spans the residuals. Truss W, Cross L and Short C carry four targets and seven residuals; Panel P carries two and three. The leading assignments use seven actuators on Truss W, three on Panel P and seven on Short C, each at the residual count, and four on Cross L, the ceiling of its bracket and three below the residual count. The two cases within tolerance are two of the three at the count; Cross L, held below it, returned the least-squares compromise the count predicts, at 11.41 mm mean; Short C, at the count, still fell short, and its error concentrates in one node at the end of the chain, the one the short-curve correction leaves without a reachable position.

7.3. The effect of released members

This section addresses RQ2: Table 7.5 reports what a release budget buys in terms of actuation. For each structure carrying degrees of self-stress, searches were run at otherwise identical settings under the procedure of Section 6.3.1: one without a release budget, and one at each ceiling up to the structure's own s_0, which for Panel P and Short C is a single search at a ceiling of one and for Cross L three. Truss W carries no budget and is therefore absent from this comparison, which is itself the baseline against which the other three are read.

Table 7.5: Best assignment per structure without a release budget and with the release ceiling at s_0. Values in millimetres.

Case	s_0	Release ceiling	n_a	Max gap residual	Max d_t	Mean d_t
Panel P	1	0	No assignment passed the usefulness gate			
Panel P		1	3	Exact placement	∼0	∼0
Cross L	3	0	No assignment passed the two-test			
Cross L		3	4	6.8867	14.7674	11.4067
Short C	1	0	No assignment passed the two-test			
Short C		1	7	1.3354	9.7852	4.6956

With no release budget none of the three structures returned a valid assignment, but the candidate sets emptied at different stages. On Cross L and Short C no candidate passed the two-test, from admissible sets of 1,820 and 94,183 respectively. On Panel P 15 of 1,940 admissible candidates

passed the two-test and all 15 were then rejected at *L*1.5: actuation alone produced mechanisms on that structure, but none whose available motion reached the target nodes Table 7.7.

Raising the ceiling to one release changed the outcome on all three. Panel P admitted 22,050 candidates, of which 16,170 passed the two-test and 15,147 survived *L*1.5 to be evaluated in full. Cross L admitted 22,295 candidates and retained 50 through both gates, of which 25 were evaluated. Short C admitted 1.197×10^6 candidates and evaluated within the given time budget 576,528. On Cross L the further released budget ceilings of two and three admitted 105,105 and 448,045 candidates and evaluated 3,201 and 63,217; across all three ceilings the count entering *L*2 was close to half the count leaving *L*1.5, a ratio consistent with a single reflection in that structure.

Cross L is the one structure on which the ceiling can be compared with other variables fixed, since it was searched at every ceiling from one to its own $s_0 = 3$ (Table 7.6). The actuator count does not move, staying at four throughout, which is the ceiling of the bracket; three-actuator candidates exist in each population (Sweep B, Section 7.4) and none was the best-placing. The placement error, meanwhile, falls at every step, from a mean of 41.55 mm and a maximum of 61.61 mm at one release to 26.15 and 52.78 mm at two and 11.41 and 14.77 mm at three. Because each ceiling's population contains those of the lower ceilings, the best at a higher ceiling can only equal or improve on the one below and did so by spending the additional release. The release sets are not nested: the single release at ceiling one (L14) is not retained at ceiling two (L8, L10), so the search re-chooses where to spend the budget rather than adding to its previous choice.

Table 7.6: Cross L: Rank-1 assignment at each release ceiling. Values in millimetres.

Release ceiling	Rank-1	n_a	Releases used	Mean d_t	Max d_t
0	No assignment passed the two-test				
1	`affafffaaffffp`	4	1 (L14)	41.55	61.61
2	`ffaafafapfpffff`	4	2 (L8, L10)	26.15	52.78
3	`fffffaffpppaaaf`	4	3 (L8, L9, L10)	11.41	14.77

Across structures, the actuator count of the best assignment does not fall as the ceiling rises. Panel P reached a valid assignment with three actuators at a ceiling of one, Cross L with four at a ceiling of three, and Short C with seven at a ceiling of one. The three structures differ in member count, target count and target curve as well as in s_0, so no relation between release budget and actuator count is drawn from the comparison across them.

Table 7.7: Candidates present at each stage of the funnel, per release ceiling, for the three structures carrying self-stress.

Release ceiling	Panel P ($s_0 = 1$)	Cross L ($s_0 = 3$)	Short C ($s_0 = 1$)	Step
0	1,940	1,820	94,183	*L*0
	15	0	0	*L*1
	0	0	0	*L*1.5
	0	0	0	*L*2
1	23,990	22,295	1.291 $\times 10^6$	*L*0
	16,185	50	1.103 $\times 10^6$	*L*1
	15,147	50	1.103 $\times 10^6$	*L*1.5
	15,147	25	576528	*L*2
2	–	125,580	–	*L*0
	–	6,420	–	*L*1
	–	6,375	–	*L*1.5
	–	3,201	–	*L*2
3	–	448,045	–	*L*0
	–	126,590	–	*L*1
	–	126,320	–	*L*1.5
	–	63,217	–	*L*2

7.4. The effect of design preferences

RQ3 names two different parameters that directly influence the output of the algorithm, and they were varied independently across one frozen candidate population under the procedure of Section 6.3.2. Sweep A varies the tolerance and the aggregation exponent with the weights at their default; Sweep B varies the weight vector with tolerance and exponent held fixed. Both are read from one frozen candidate population, so the Rank-1 assignment changes between rows only because the row's own parameter changed.

Sweep A

Table 7.8: Sweep A: Rank-1 assignment across the tolerance τ and the aggregation exponent q, with the weights at their default. Values in millimetres.

Case	τ	q	Rank-1	Mean d_t	Max d_t	Score
Truss W	0.1	∞	`fffffaffffaafffa`	70.91	179.05	37.26
	0.1	2	`fafffffafaaffaafa`	54.84	123.15	40.50
	1	∞	`fffaaaaafaffafff`	0.51	0.68	64.77
	1	2	`fafaaaaafaffafff`	0.24	0.49	72.75
	5	∞	`affafaaafffaafff`	2.19	4.88	80.69
	5	2	`fffaaaaafffaafff`	3.68	9.13	80.85
Panel P	0.1	∞	`fffffafapffffa`	Exact placement		88.44
	0.1	2				
	1	∞				
	1	2				
	5	∞	`fffffffapffffa`	3.27	4.92	94.26
	5	2	`fffffffapffafff`	4.20	5.36	94.32
Cross L	0.1	∞	`afffffffppapffa`	57.74	64.26	39.98
	0.1	2				
	1	∞				
	1	2				
	5	∞	`faffffpapfpfaff`	31.46	57.92	54.58
	5	2				
Short C	0.1	∞	`affffafafffffffpfaf`	32.70	67.88	54.12
	0.1	2				
	1	∞	`fpffffffaafffffffafaf`	35.29	67.95	56.82
	1	2				
	5	∞				
	5	2	`ffaafffpfaffafffaaa`	4.69	9.78	70.51

In Sweep A the tolerance is the parameter that moves the result. On Truss W the Rank-1 assignment changes at every step of τ, and placement follows it: at τ = 0.1 mm the leading candidate places at a mean of 70.91 mm, at τ = 1 mm at 0.51 mm, and at τ = 5 mm at 2.19 mm. The same pattern can be partially found on the other three structures, with Short C improving from a mean of 32.70 mm at τ = 0.1 mm to 4.69 mm at τ = 5 mm. Tightening the tolerance does not directly tighten the placement achieved due to the subscore function in Section 5.6. Above τ the deadband decays as $\exp(-(e-\tau)/\tau)$, so at τ = 0.1 mm a candidate placing at 70.91 mm and one placing at 0.51 mm both score effectively zero on placement. Under the weights in force, placement and spacing together carry 0.6 of the score, and both are dead at that tolerance unless a candidate's gap residuals happen to fall within a fraction of a millimetre; the rank is therefore decided by the count, effort and path subscores, which is why the τ = 0.1 mm scores on Truss W and Cross L sit near 40, the ceiling those three subscores can reach, against 80 and above at τ = 5 mm. The same flattening acts from the other side: at τ = 5 mm every candidate with $\bar{e}_q$ below 5 mm scores 1 on placement, so the leading assignment at τ = 5 mm places worse than the one at τ = 1 mm because placement no longer separates them and the other subscores

decide again. Only where the tolerance falls within the range the mechanism can reach, as at τ = 1 mm on Truss W, does the subscore separate the candidates that reach it from those that do not.

Sweep B

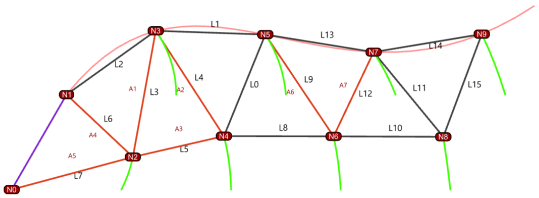
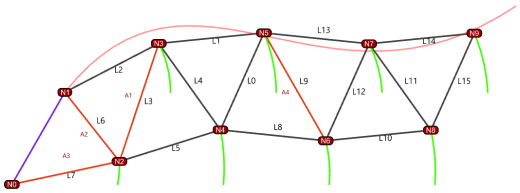
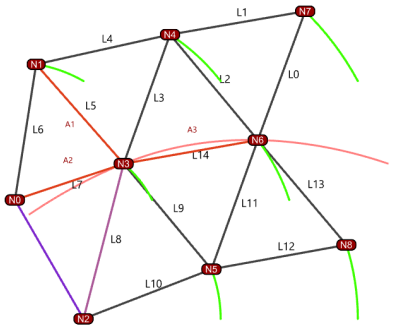
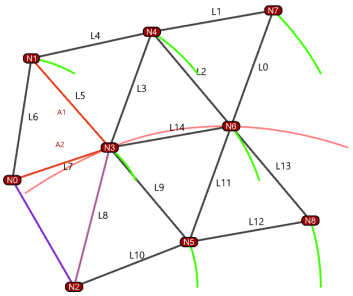
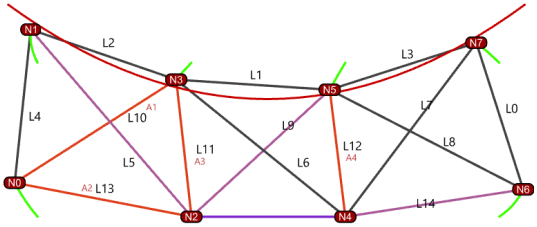
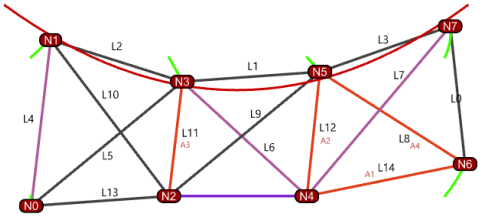
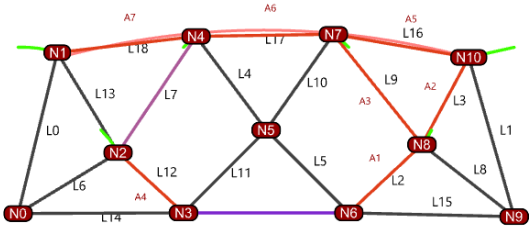
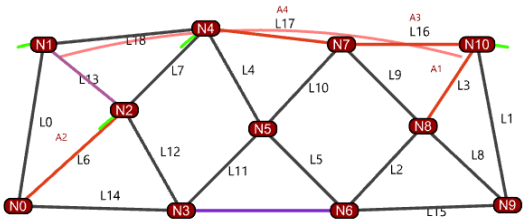
Table 7.9: Sweep B: Rank-1 assignment across weight vectors, with τ = 5 and q = 2 held fixed. Values in millimetres.

Case	Weights (p/s/c/e/pa)	Rank-1	n_a	Mean d_t	Max d_t	Score
Truss W	1/0/0/0/0	`fffaaaaafaffafff`	7	0.51	0.68	100
	0.5/0/0.5/0/0	`affaffaafaffafff`	6	4.38	12.04	63.88
	0.333/0/0/0.667/0	`fffffaffaaaaafaa`	8	69.35	179.05	54.66
	0/0/1/0/0	`fffaffaafafffff`	4	19.32	39.75	100
Panel P	1/0/0/0/0	`fffffafapffffa`	3	Exact placement		100
	0.4/0.2/0.15/0.1/0.15	`fffffffapffafff`	2	4.20	5.36	94.32
	0.333/0/0/0.667/0	`fafffffapffafaf`	4	4.20	5.36	81.07
	0/0/1/0/0	`fffffafapffffff`	2	0.39	0.40	100
Cross L	1/0/0/0/0	`fffffaffpppaaaf`	4	11.41	14.77	25.41
	0.4/0.2/0.15/0.1/0.15	`fafffffpapfpfaff`	3	31.46	57.92	54.58
	0.5/0/0.5/0/0	`fffapfpfpffaaff`	3	15.99	24.13	54.36
	0/0/1/0/0	`fffapfpfpffaaff`	3	15.99	24.13	100
Short C	1/0/0/0/0	`ffaafffpfaffafffaaa`	7	4.69	9.78	85.33
	0.4/0.2/0.15/0.1/0.15	`ffaafffpfaffafffaaa`	7	4.69	9.78	70.51
	0.333/0/0/0.667/0	`afpaffaafffffafffff`	5	51.63	84.56	66.64
	0/0/1/0/0	`fffaffffpffffaffafa`	4	25.53	29.87	100

In Sweep B the weight vector changes both the assignment and the actuator count. Moving from pure placement to pure count weighting, the count falls from seven to four on Truss W, from three to two on Panel P, from seven to four on Short C and from four to three on Cross L, with the mean placement error rising from 0.51 mm to 19.32 mm on Truss W and from 4.69 mm to 25.53 mm on Short C. On the first three structures the count stops at the number of target nodes, and on Cross L at the floor of its bracket, below its four targets. Both end points are set by the count subscore rather than by the structures: every assignment with at most n_t actuators scores 1 (Section 5.6), so under pure count weighting all of them tie at 100, and the assignment reported is the one the search returned first among them. Pure count weighting therefore states that n_t actuators are enough, and does not discriminate below them. A score is a statement about a preference and is not comparable across weight vectors; the default vector sits between the two extremes, and on Short C it selects the same assignment as pure placement weighting while scoring it 70.51 rather than 85.33.

Two rows in Sweep B derive their scores from the saturation of the subscores rather than from the trade between count and placement. On Cross L the pure-placement vector returns four actuators at a mean of 11.40 mm and a score of 25.41, the lowest in either sweep: every candidate lies outside the tolerance, so all score low on placement and pure placement weighting leaves nothing else to score, while the default vector, which also rewards the saved actuator, returns three actuators at 31.46 mm and a higher score. On Panel P the pure-count vector returns two actuators at a mean of 0.39 mm, better placement than the default vector achieves with the same count at 4.20 mm. Both candidates sit inside the 5 mm deadband, where the default vector scores them alike on placement and decides between them on spacing, effort and path, while under pure count weighting the 0.39 mm candidate is one of many tied at 100. Table 7.10 shows some of the configurations these rows produce.

Table 7.10: Rank-1 configurations of each case under two weight vectors (p/s/c/e/pa).

Case	Weights	Rank-1	Weights	Rank-1
Truss W	1/0/0/0/0		0/0/1/0/0	
Panel P	1/0/0/0/0		0/0/1/0/0	
Cross L	1/0/0/0/0		1/1/0/0/0	
Short C	1/0/0/0/0		1/0/1/0/0	

7.5. From computational to physical

This section addresses RQ4. The baseline run (Section 4.6) showed that the demonstrator executes a hand-planned transformation; here, an inert starting lattice instead receives a role for each member and a travel percentage for each actuator from a computed search, and is then driven through the transformation the framework produced. Truss W and Panel P were processed both with the target curve shown in the previous sections and with one additional target: For Truss W a convex curve with no flection; for Panel P a curve with an inverse curvature compared to the first. The second Panel P search was run with a wider actuator bracket ($k_{\min} \dots k_{\max} = 3 \dots 8$) and a tolerance of τ = 5 mm. The resulting four assignments were applied physically and actuated through the control pipeline, and each run was recorded under the criteria of Section 6.3.3. The Mean d_t, Max d_t columns in the tables that follow are the computed values of the selected candidate, restated from the search output.

Table 7.11: Truss W, Curve 1: computed (left) and physical (right) configuration before (top) and after (middle) actuation, and the physical result overlaid on the computed one (bottom). Values in millimetres.

Case	τ	q	Rank-1	n_a	Mean d_t (comp.)	Max d_t (comp.)	Executed?
Truss W	5	2	`fffaaaaafaffafff`	7	0.51	0.68	Yes

Curve 1

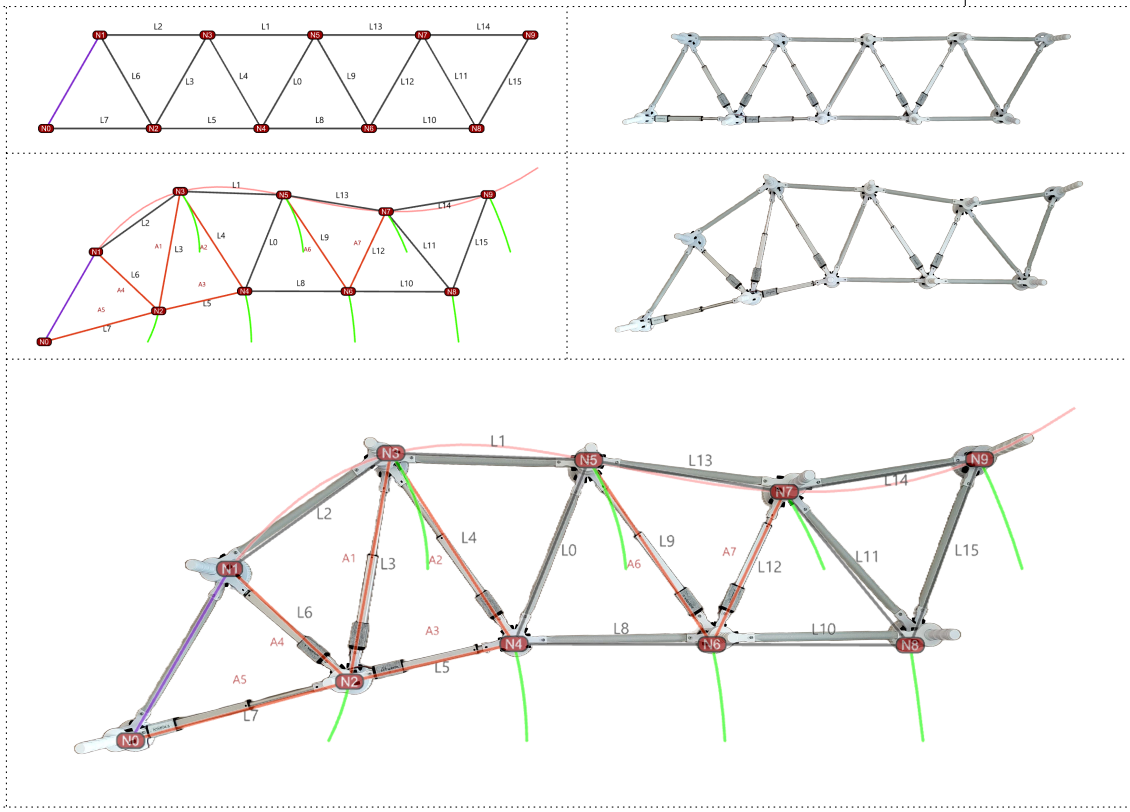
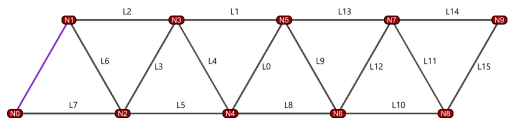


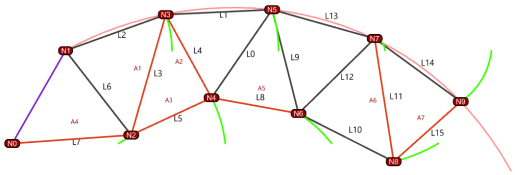
Table 7.12: Truss W, Curve 2: computed (left) and physical (right) configuration before (top) and after (middle) actuation, and the physical result overlaid on the computed one (bottom). Values in millimetres.

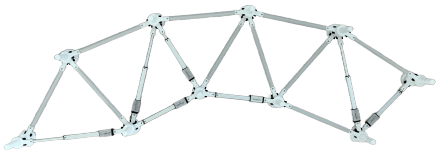
Case	τ	q	Rank-1	n_a	Mean d_t (comp.)	Max d_t (comp.)	Executed?
Truss W	5	2	`fffaaafaaffafffa`	7	0.62	1.01	Yes

Curve 2









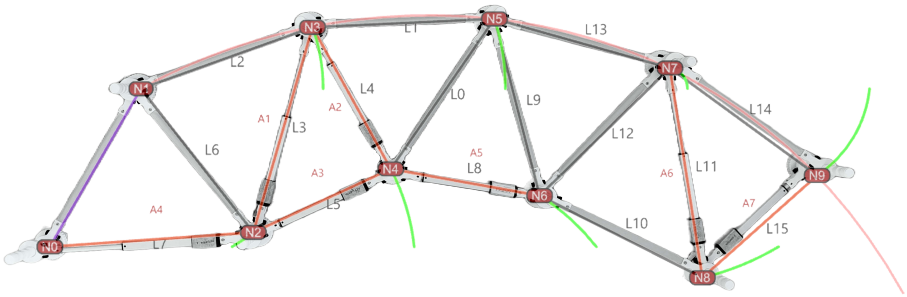


Table 7.13: Panel P, Curve 1: computed (left) and physical (right) configuration before (top) and after (middle) actuation, and the physical result overlaid on the computed one (bottom). Values in millimitres

Case	τ	q	Rank-1	n_a	Mean d_t (comp.)	Max d_t (comp.)	Executed?
Panel P	5	2	`fffffafapffffa`	3	Exact placement		Yes

Curve 1

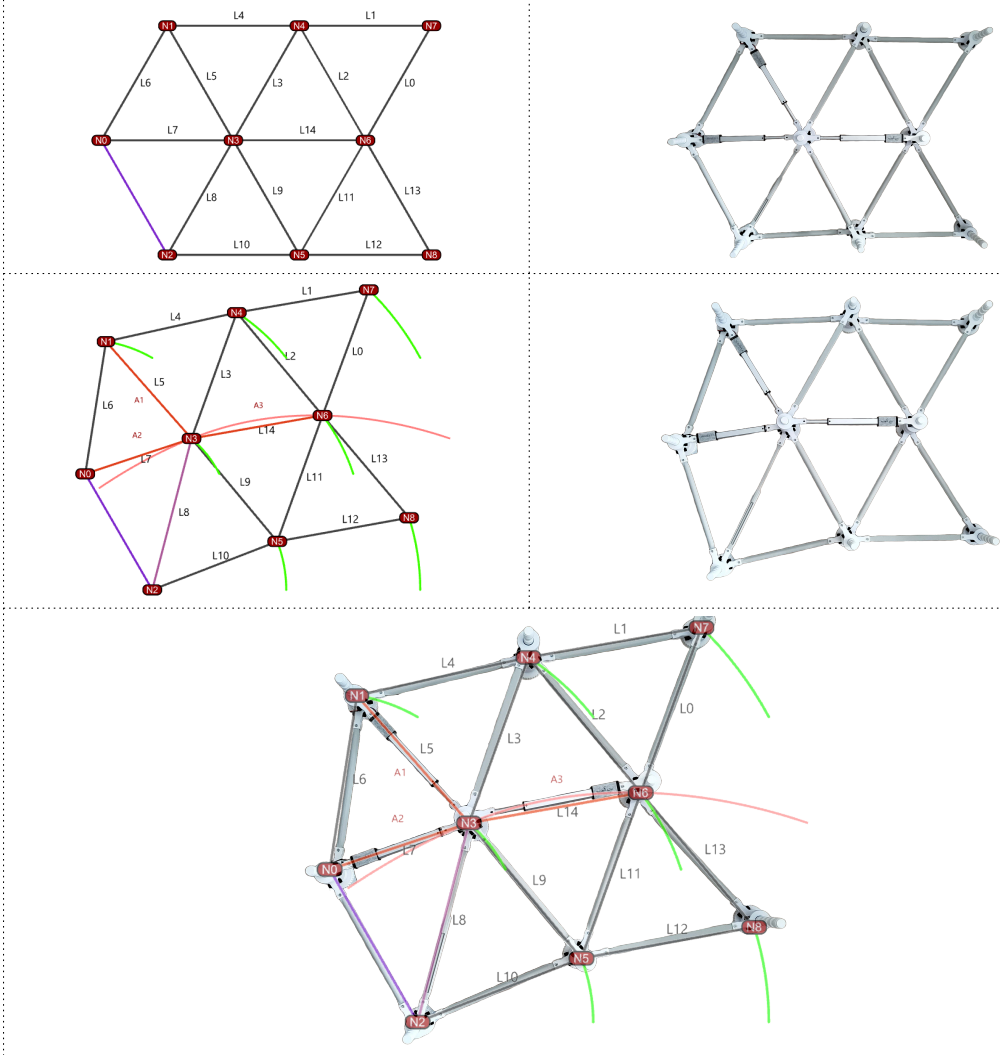
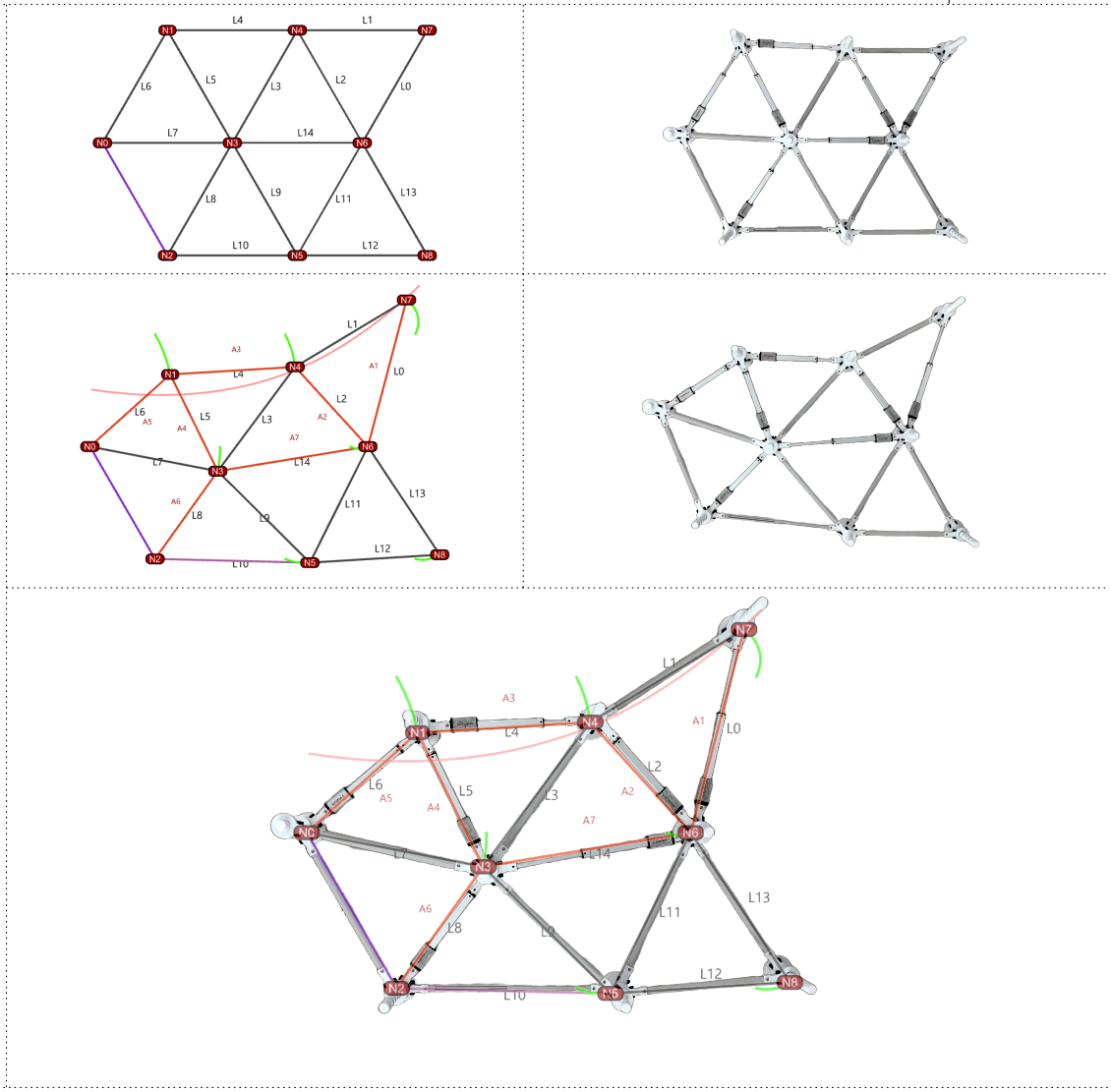


Table 7.14: Panel P, Curve 2: computed (left) and physical (right) configuration before (top) and after (middle) actuation, and the physical result overlaid on the computed one (bottom). Values in millimetres.

Case	τ	q	Rank-1	n_a	Mean d_t (comp.)	Max d_t (comp.)	Executed?
Panel P	5	2	`affaaaapffaafff`	7	21.45	24.13	Yes

Curve 2



All four assignments were carried out on the demonstrators, and in each case the structure moved under actuation, remained driven throughout the travel, and arrived at a configuration corresponding to the computed one on visual overlay, as the photographs accompanying each block show. None jammed or stalled short of its prescribed travel. The two Panel P assignments are the first in which the released role was exercised physically rather than computationally: the telescopic released member was driven by the forces supplied by the active members, as the role assumes, and in both cases it accommodated the motion without leaving its bounds.

The results therefore establish that the output of the search can be translated into a physical actuation without modification. What the algorithm returns (a role per member and a percentage per actuator) is sufficient to build and drive the structure, across assignments ranging from three actuators to seven. No further interpretation, hand-tuning, or case-specific adjustment was required between the computed result and the executed transformation. What the runs do not establish is accuracy: the realised node positions were not measured, so how closely the physical structure approaches the computed geometry, or the target curve, is not quantified here.

7.6. Answers to the research questions

The table collects the evidence bearing on each question and states what it supports.

Table 7.15: Evidence and answer for each research question.

Research question	Evidence	Answer
RQ1: Which assignments constitute a driven mechanism able to move the targets, and how does actuator placement determine the motion available	Section 7.1, Table 7.1, heatmaps in Table 7.2	The admissible set accounted for between 0.09% and 3.12% of the raw 3^n space across the four cases, and the verified set for between 0.05% and 0.44%. Almost all of that reduction is obtained by generating the admissible set from the brackets rather than by filtering, since the gates rejected nothing at all on Truss W. Where the structure carries self-stress, the gates do filter: the two-test rejected 8% on Short C, 27% on Panel P and 72% on Cross L. Placement, since the enumeration makes every member equally frequent across the admissible set, is legible only among the best-placing candidates: actuation concentrates on the members at the grounded nodes on Truss W, Panel P and Cross L, and on the chord between the targets on Short C, whose curve is shorter than the chain (Table 7.2).
RQ2: How the integration of released members changes the actuation required	Section 7.3, Tables 7.5 to 7.7	With the release ceiling at zero none of the three structures carrying self-stress returned a valid assignment. On Cross L and Short C no candidate passed the two-test; on Panel P fifteen did but then all failed at the usefulness gate. Both outcomes follow the release-free form of Equation 3.4: without releases an actuator is effective only on a member that carries no self-stress. Every live member of Short C carries its state of self-stress, at most two members of Cross L are free of it, and the fifteen Panel P mechanisms are the combinations of its four free members. Within the scope of independent actuation, releases are therefore a precondition for adaptation rather than an economy available to it. Raising the release budget produced valid assignments on all three, and on Cross L, where the ceiling could be raised in steps, the actuator count stayed at four while the mean placement error fell from 41.55 mm to 11.41 mm, with budget buying reach.
RQ3: How acceptable closeness and stated preference change which assignment is selected and which configuration is reached	Section 7.4, Sweeps A and B (Tables 7.8 and 7.9)	Both change it. The tolerance moves the leading assignment at every step tested, and tightening it does not tighten the placement achieved: on Truss W the mean placement error at τ = 0.1 mm was 70.91 mm against 0.51 mm at τ = 1 mm because far outside the tolerance the placement subscore is alike and the count, effort and path subscores decide the rank. The weight vector moves both assignment and actuator count, which fell under the number of target nodes on Truss W (7→4), Panel P (3→2) and Short C (7→4), and to the bracket floor on Cross L (4→3), with mean placement error rising correspondingly. That end point is set by the count subscore, which scores every assignment at or below n_t actuators alike, so the selection among those assignments is not made by the score.
RQ4: Whether the prescribed assignments can be executed physically	Section 7.5, Tables 7.11 to 7.14	Four assignments were built and driven on two structures, spanning three to seven actuators and including two carrying a released member. All four executed: the structures moved under actuation, remained driven throughout the travel, and reached configurations corresponding to the computed ones on visual overlay, with no binding or stalling. The output of the search is therefore executable without interpretation or hand-tuning. The realised geometry was not measured and compared against the computed values, so the result is one of executability rather than accuracy.

8

Discussion

8.1. Central finding

A decisive variable in adapting a given lattice is which members are released, not only where actuators are placed.

On all three structures carrying self-stress, a search permitted to actuate but not to release returned no valid assignment. This is independent from the case set and grounded in the release-free form of Equation 3.4: without releases, an actuator passes the two-test only on a member that carries no state of self-stress, at any actuator count. The counts of Table 7.7 make this visible. On Short C none of 94,183 candidates passed, including all nineteen single-actuator ones, so every live member carries its state of self-stress. On Cross L, searched from three actuators upward, none of 1,820 passed, so at most two of its members are free of self-stress. On Panel P exactly fifteen passed, the number of non-empty combinations of four members within a bracket of one to four actuators: four members carry no self-stress, and none of their combinations moved the targets. With one release permitted, the condition predicts that every assignment releasing one of the eleven members carrying Panel P's state passes and every other fails, giving $11 \times 1{,}470 = 16{,}170$ candidates, the count recorded at *L*1. Useful mobility appears once releases have removed every state of self-stress that loads an actuator: the release makes the actuation effective rather than replacing it.

The finding holds within the scope of the two-test, which requires every actuator to add an independent degree of freedom (Section 5.4.2). A redundant frame can also be driven by redundant actuation, with actuators on members that carry self-stress and commanded compatibly so that the self-stress is never engaged. That route trades releases for coordinated control and is unavailable to open-loop actuation control. Within this scope, the central finding behaves as a condition: an actuator is effective only on a member that carries no self-stress once the releases are made, and in a frame where every member that is able to move the targets carries self-stress, releases are the only route to meaningful actuation.

Truss W, where releases are impossible since $s_0 = 0$, neither gate rejected a single candidate out of the enumerated set: every admissible assignment was a valid mechanism whose motion reached the targets. The question of which assignment to use collapses therefore to placement quality alone, as validity could not be questioned. Gates *L*1 and *L*1.5 can be considered useful only where the decision of which member to release exists.

Where actuation belongs, the second half of RQ1, follows a pattern the heatmaps of Section 7.1 make visible. On Truss W, Panel P and Cross L the favored members are those at the grounded nodes: L6 and L7, with L3 beside them, on Truss W; L7 and L8 on Panel P; the interior verticals and middle-bay diagonals on Cross L. These are the first links of every chain from the ground to the targets, so a change in their length moves every node beyond them, whereas a member near the free end moves only the nodes it touches. Short C departs on both counts: its top chord is favored and most members at its grounded nodes are not. Its curve is shorter than the chain of targets, so the end targets reach

it only if the chain between them shortens, and the chord is the most direct way to shorten it. Where actuation belongs is therefore a property of the target as well as of the topology.

A direct consequence for how an adaptive lattice is designed can be distilled. A designer choosing a topology for a structure that must move is choosing, implicitly, which members carry self-stress, and that decides which members can take an actuator without a release and how many releases each placement costs, before any question of placement quality arises. It is also the more expensive decision computationally, since the release ceiling enters the admissible set as an exponent while the member count enters only as a base (Section 5.4.1).

8.2. What the search reduces, and what it does not

Three claims follow from the funnel results:

- **Most of the reduction is obtained by generation.** The admissible set is constructed directly, by selecting the active members and then the passive ones only in the numbers the actuator and release bounds permit (Section 5.4.1), instead of by listing all 3^n assignments and discarding those outside the search settings. That construction alone reduced the space from 0.09% to 3.12% of 3^n across the four cases, without testing a single candidate. The gates act afterwards on what was constructed, removing assignments that are admissible but are not valid or useful mechanisms. The ordering matters because the two steps differ in both computational cost and effect: construction of the admissible step involves no calculations at all, while every gate requires one decomposition per candidate, so the step that does most of the reducing is also the one that costs least.
- **The gates are not redundant with one another, but they act only when self-stress exists.** On Panel P the two-test rejected 27% and the usefulness gate a further 6%; while on Cross L structure the two-test rejected 72% and the symmetry quotient then halved what survived. On Truss W neither gate rejected a candidate, and on Short C the two-test rejected 8% and the usefulness gate nothing. Removing either gate would therefore change nothing on the isostatic case and little on Short C, while deciding the outcome on Panel P and Cross L, and which side a structure falls on is determined by s_0 rather than by size. The symmetry quotient fired on one case of four, exactly halving it, which is consistent with a single reflection; the absence of a reduction elsewhere reflects the structures rather than the method.
- **The reduction has an edge.** The enumeration of the admissible set is multiplicative on a set that grows polynomially in the member count at fixed brackets, with the exponents set by the actuator and release ceilings. Short C is the one where the limit became visible: its 1.103×10^6 valid candidates could not all be evaluated within the time budget, and the 576,528 that were represent a truncation. The framework does not degrade gracefully past that point: the enumeration must be materialised before any gate can act on it, so beyond some combination of member count and ceilings it stops rather than slows.

That last point bears directly on the central finding. An assignment needs at least as many releases as there are independent states of self-stress loading its actuators (Section 3.3), so in a heavily redundant frame the share of valid assignments rises with the release level: on Cross L, 50 of 22,295 admissible candidates passed the two-test at a ceiling of one, 6,420 of 105,105 at two and 126,590 of 448,045 at three (Table 7.7). The enumeration reaches the high levels last, because the inner sum over release counts rises steeply and its largest terms are the highest. The region holding most valid assignments and the most expensive region of the enumeration coincide, so a search restricted to low release levels reports few or none.

The actuator count of a leading assignment is legible against the residual count of its target, $2n_t - 1$ for n_t target nodes on a curve long enough to hold them (Section 5.5). Where the search was permitted to reach that count it did so under placement weighting: seven actuators for seven residuals on Truss W and Short C, three for three on Panel P. Where it was not, on Cross L with a bracket capped at four, the best assignment sat at the cap and placed at a mean of 11.41 mm, and raising the release budget improved that figure without changing the count. The weight sweeps read the same way from the other side. Under pure count weighting Truss W dropped to four actuators against seven residuals and its mean error rose from 0.51 mm to 19.32 mm; Panel P dropped to two against three and rose only from

below reporting precision to 0.39 mm, since a target close to the drawn configuration leaves little for the missing actuator to do. A similar dependence is reported by Sachse et al. (2021), where prescribing more of the target shape required more actuators.

Two consequences follow for how the framework is used. A bracket whose $k_{\max}$ lies below $2n_t - 1$ cannot generically reach tolerance, and a search meant to test reachability should receive an actuator cap at least equal to the residual count. The actuator-count subscore of Section 5.6 gives full marks at n_t actuators, one per target, which is below the residual count $2n_t - 1$ whenever more than one target is placed: pure count weighting therefore drives the selection to a count at which the targets cannot generically reach tolerance while holding the spacing, which is why it trades placement steeply on Truss W.

8.3. Redundancy as a precondition

Section 3.3 states two things about releases: that s_0 caps how many a frame can spend while remaining rigid with its actuators held, and that every independent state of self-stress loading an actuator must be released before that actuator yields motion. The results confirm both, and on the structures tested the budget resulted as a requirement.

With no release permitted, Cross L and Short C produced no mechanism, and Panel P produced mechanisms only from the four members that carry no self-stress, none of which moved the targets (Section 8.1). In these frames the release budget is therefore not a cheaper route to mobility but the only one, and a release does not substitute for an actuator: it removes the constraint that would otherwise absorb the actuator's travel.

What the release budget buys can be answered in two ways. Across structures the actuator count of the best assignment does not fall as the ceiling rises, but the three differ in member count, targets and target curve as well as in s_0, so that comparison is uncontrolled. Within a single structure it can be controlled, and Cross L provides the case. With the ceiling raised from one to three as the only variable, the actuator count stayed at four while the mean placement error fell from 41.55 mm to 11.41 mm and the maximum from 61.61 mm to 14.77 mm. On this structure a release bought reach: the trade of one release for one state of self-stress appears not as a saving in actuators but as motion that the same number of actuators could not otherwise direct at the targets. At ceilings of two and three the best-placing assignment used every release permitted. Equation 3.4 does not require this, since validity needs only as many releases as the states of self-stress loading the actuators, and valid assignments with fewer releases existed at each ceiling; the full budget was spent because it gave the placement more freedom, not because validity demanded it. Without the released role, the redundancy must be taken up by actuators instead, which is the situation of the method reviewed in Section 2.2.

8.4. Role collapsing: one member for all

The framework assigns to every member one of three roles (active, fixed, or passive) and treats them as three different physical elements: an actuator, a rigid member and a telescopic/passive strut. A natural development for post-deployment adaptation is a single physical member that can be any of the three, switched between roles rather than swapped as the demonstrators here require. This matters because the current alternatives force a choice between scale and reconfigurability. At the centimetre scale the demonstrator targets, a dedicated actuator generates motion but adds weight and wiring that scale badly once a lattice has hundreds of nodes to drive independently. A member that is only telescopic, or only rigid, is cheap but cannot generate motion of its own. A member that could actively drive its own length, then hold that length rigidly and otherwise sit as a free-length passive strut, would let one component cover all three roles the search assigns.

The idea of one member switching between driving and holding has direct precedent in research, where members double as actuators and load paths within a single reconfigurable element (Senatore et al., 2019). What is proposed here differs in aim: while that work optimises actuator layout for energy under load, this framework looks for kinematic roles. The shared idea is a member that both moves and locks, and their hardware is a close analogue to a fused element at this scale. The "active then fixed" sequence (a structure morphs into a target shape, then holds it) is also the operating principle of the shape-morphing truss structures of Sofla et al. (2009) reviewed in Section 2.1.3. Those structures

morph between designed shapes rather than conforming toward a given target curve, which is the step this framework's inverse search adds, but the duality of roles a member carries is identical.

The requirement such a member must satisfy is that it should be simple and cheap enough that assigning it to the fixed role is not a waste. The adaptation algorithm assigns most members the fixed role, so a fused member that is expensive or heavy would make every fixed placement counterproductive, and the efficiency of the solutions the whole framework is built to find would be lost at the hardware level. The goal should therefore be not a maximally capable member but a minimally capable one: just enough mechanism to switch between driving, locking and releasing, and no more, so that a lattice built entirely from these members pays little for the ones that end up holding still. Only then does the framework's output map onto a structure a designer can build from a single part type and reconfigure after deployment.

8.5. Applications

The framework is applied here to two examples. The first is a wing section: a lattice rib fitted to one airfoil profile and actuated until it reaches a second. The second is a bed frame: one planar frame actuated into a reclined posture matched to an occupant profile. The two cases are used to show what the framework returns on concrete target geometries; they are not developed into finished engineering designs.

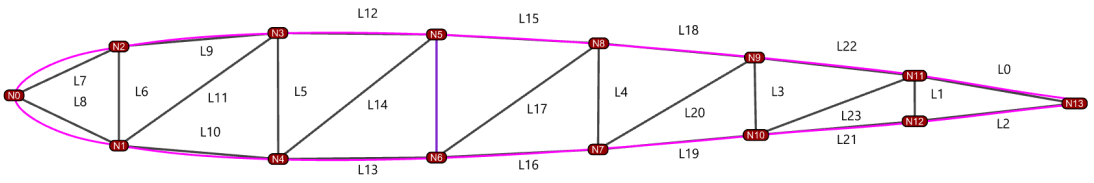
8.5.1. Application case: section morphing of a wing rib

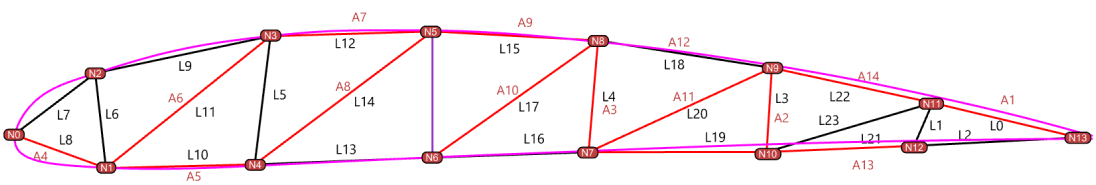
Discretely assembled lattices have already been applied to wing structures, with research concentrating on elastic shape change rather than kinematic reconfiguration. Jenett et al. (2017) assembled a wing from modular composite lattice parts on a NACA 0012 profile and produced continuous span-wise twist through a single torque tube. Cramer et al. (2019) approached an aircraft built from two building-block types, spatially programming the elastic deformation of the primary structure to improve aerodynamic efficiency and roll control. Wang et al. (2022) introduced distributed actuation, embedding extending and contracting active cells in a heterogeneous lattice and searching over cell sequence and actuator inputs to reach a target contour within a maximum deviation of 3 mm. In each case the shape change is elastic and the design variable is the spatial arrangement of cells of differing stiffness, rather than the actuation scheme of a frame whose cells are already fixed.

Applied to a wing section, the framework reaches the target profile kinematically instead, by selecting which members of a fixed frame are actuated. A planar rib fitted to a NACA 0012 profile was given the corresponding node positions on a NACA 4412 profile as its target, and the resulting Rank-1 assignment actuates 14 members and leaves the remaining beams fixed, reaching the target with a mean deviation of 2.79 mm and a maximum of 6.62 mm. Cells, member topology and node count are identical in both states, so the frame that holds the symmetric profile can also hold the cambered one.

Table 8.1: Wing rib: Rank-1 assignment for the NACA 0012 to NACA 4412 morph, with the rib before (top) and after (bottom) actuation. Values in millimetres.

Case	τ	q	Rank-1	n_a	Mean d_t	Max d_t	Score
Wing	5	2	`affaaffaaffaafaafafaaaaff`	14	2.79	6.62	75.23





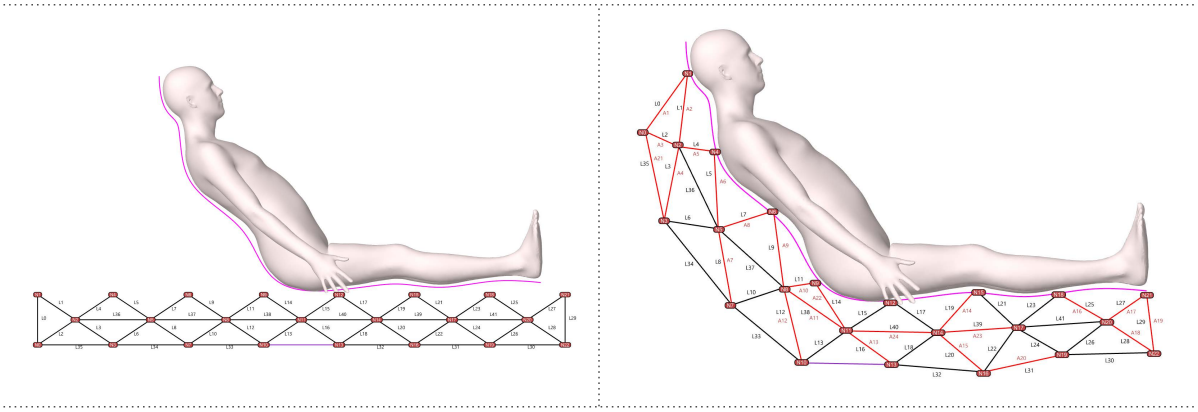
8.5.2. Application case: posture fitting in a mobile bed frame

Long-term care beds are built around a fixed articulation geometry. A profiling bed divides its mattress platform into a small number of sections, whose pivot axes sit at set positions along a frame of standard length, so the occupant is fitted to the bed rather than the bed to the occupant. Where manufacturers have addressed body size directly, they have done so by building a different bed: one such system specifies four standard frame lengths and measures each user's head-to-hip and hip-to-knee distances to fix the pivot positions at manufacture (Hillrom, 2026). The fit between body and mechanism is therefore decided before the bed is delivered.

Applied to a reclining bed frame, the framework keeps the structure constant and varies only the actuation. A planar truss, flat in its initial state, was given the reclined posture of an occupant as its target, sampled as a continuous contour along the back, seat and legs rather than as a set of segment angles. The highest-ranked assignment actuates 25 members and follows the contour to a mean deviation of 3.38 mm and a maximum of 11.25 mm, at a tolerance of 10 mm; as for the wing, the actuator count is the price of the fit, and a count-weighted re-score would trade placement for fewer actuators (Section 7.4).

Table 8.2: Bed frame: Rank-1 assignment for the reclined contour, with the frame before (left) and after (right) actuation. Values in millimetres.

Case	τ	q	Rank-1	n_a	Mean d_t	Max d_t	Score
Bed	10	2	`aaaaaafaaafaafffaffaaf` `fafafaaafafffaffaaafa`	25	3.38	11.25	100



8.6. Limitations

The framework reasons about motion alone. It calculates where to drive the structure, but never whether the structure can hold itself once it arrives. No member sizing, stress, buckling, self-weight, external load or actuator force enters the computational workflow at any point, so an assignment scoring 100 for placement may be unable to stand under its own weight. This is the widest gap between the tool as it stands and a usable engineering method, and closing it is the first item of future work (Section 9.2).

The rigidity analysis is first order only. The tests read the rank of a linearised matrix, and a structure that is a mechanism to first order can be stiffened by a state of self-stress and be genuinely rigid at second order.

Every verdict belongs to the geometry as drawn. The matrix is evaluated at the coordinates in the Rhino model rather than on a normalised topology. A verdict therefore belongs to those coordinates and no others: the same topology drawn at slightly different positions must be re-evaluated.

The search is bounded by enumeration. Every gate acts on candidates that have already been constructed, so the admissible set must be generated in full before any reduction takes effect. Its size grows polynomially with the member count and far faster with the release ceiling, and beyond some combinations of the two the framework cannot run at all rather than running slowly. The limit was reached within the case set: on the short-curve structure only 576,528 of the 1.103×10^6 valid candidates were evaluated within the time budget (Table 7.7), so its reported results are the best of a

subset rather than of the full valid set. The constraint is sharpest in heavily redundant frames, where the valid assignments sit at the highest release levels and are therefore the last to be reached.

The returned rank is not always the closest placement or the highest score. The Rank-1 rows reported in Chapter 7 are the candidates the search returned first, and two mechanisms separate them from the best candidate in the population. The first is saturation: inside the tolerance, at or below n_t actuators and two decades above the singularity threshold the corresponding subscores reach 1, and candidates differing only in what a saturated subscore no longer reads tie exactly. Among tied candidates the scoring defines no order, so on Truss W the placement-only Rank-1 places at a mean of 0.51 mm while the same population holds a candidate at 0.24 mm (Table 7.8), and under pure count weighting the four-actuator assignment reported ties with every assignment of fewer actuators. The second was observed and not isolated: the second Panel P search run for the demonstrator returned an assignment placing at a computed mean of 21.45 mm while the stored population held candidates placing closer, found manually through a search in the stored values. Under the default weights a closer placement does not by itself imply a higher score, and the ordering applied inside the search was not traced within the time of this work.

The funnel counts (RQ1), the emptying of the candidate set without releases (RQ2) and the executability of the returned assignments (RQ4) do not depend on the ranking. The Cross L release series of RQ2 lies entirely outside the tolerance, where the placement-only score decreases strictly with the aggregate error, so ties cannot arise there. The re-scoring comparisons of RQ3 and the within-tolerance rows of Table 7.3 do depend on it. “Rank-1” should therefore be read throughout as “returned first by the search” rather than as “highest-scoring in the population” or “closest placement in the population”.

The placement subscore is informative only near the tolerance. The deadband of Section 5.6 is flat at 1 below τ and decays exponentially above it, so it separates candidates only within a few multiples of τ. Set far below what the mechanism can reach, it scores every candidate near zero and the ranking passes to the count, effort and path subscores, which is why the τ = 0.1 mm rows of Sweep A return candidates placing at tens of millimetres with scores near 40 points. The remaining three subscores carry under the weights used, and with a placement-only weighting the same setting would leave every candidate tied at zero. Set above what the best candidates can reach, it scores all of them 1 and the same transfer happens from the other side. The finding reported under RQ3, that tightening the tolerance does not tighten placement, is therefore a property of both the scoring design and the structures. A subscore that decays more slowly with error, or that normalises the error by the range present in the population, would keep placement informative at tight tolerances, and the re-scoring architecture would allow either to be applied to the stored populations without a new search.

The physical results establish executability. The four assignments of Section 7.5 were built and driven to completion, but the realised node positions were not measured, so nothing in this thesis quantifies how closely the physical structure approaches the computed geometry or the target curve.

Verification rejected nothing. The verification stage ran on every TopN = 3 candidate but rejected none (Section 7.1), so whether its singularity threshold and passive length bounds separate buildable from unbuildable assignments could not be demonstrated here.

Roles are set at assembly. The framework prescribes which members should be active, fixed or passive, and the physical structure is then reconfigured by hand to match, exchanging members at the specified indices before actuation begins, and each of the four physical runs was realised this way. This is what separates the work from reconfiguration after deployment: a structure built to one assignment cannot adopt another without being partly disassembled, so a target that changes after installation would require intervention rather than a command. Section 8.4 sets out what a member capable of switching roles would have to be, and that route is speculative.

Case-set scope. The case set was composed of four planar structures of fifteen to nineteen members. One is isostatic and the other three carry degrees of static indeterminacy (Table 7.5), so what this chapter states about heavily redundant frames rests on the rank argument of Section 3.3 and on one measurement within the case set, the Cross L series, rather than on a range of redundancies.

9

Conclusion and future work

9.1. Conclusion

This thesis asked under what conditions a pin-jointed lattice of given topology can be driven, controllably toward a prescribed target geometry. The conditions for mobility alone were already established; what this work sets out is the set of further conditions under which mobility becomes useful, targeted and buildable adaptation, and it does so by answering four questions about a lattice of fixed topology.

RQ1 asked which role assignments constitute a driven mechanism capable of moving the target nodes, and how the placement of actuation determines the motion available. Across the four cases the admissible set accounted for between 0.09% and 3.12% of the raw 3^n space, and the set carried through to verification for between 0.05% and 0.44% (Table 7.1). Nearly all that reduction comes from constructing the admissible set directly from the actuator and release bounds rather than from the gates that follow. The gates act only where the structure carries self-stress: on the isostatic Truss W neither rejected a single candidate, while on the three redundant structures the two-test rejected between 8% and 72% of what reached it. Because every member appears equally often across the admissible set, placement becomes legible only among the best-placing candidates, where it concentrates on the members at the grounded nodes, the first links of every chain to the targets, except on Short C, whose curve is shorter than the chain and where the chord between the targets is favoured instead (Table 7.2). Under placement-focused scoring the leading assignment placed every target within tolerance on Truss W and Panel P and fell outside it on Cross L and Short C at maximum placement errors of 14.77 mm and 9.79 mm (Table 7.3).

RQ2 asked how the integration of released members changes the actuation required. With no release permitted, none of the three structures carrying self-stress returned a valid assignment (Table 7.5): on Cross L and Short C no candidate passed the two-test, and on Panel P the fifteen that did produced no motion at the targets. This follows from a rank condition independently from the case set (Equation 3.4): an assignment is a driven mechanism only if its releases remove every state of self-stress that loads an actuator, so without releases only members carrying no self-stress can be actuated, and an assignment needs at least one release per independent state of self-stress loading its actuators. Within the scope of actuation, the release budget is therefore a precondition for adaptation in these structures rather than an economy available to it. A larger budget did not produce a more economical assignment in terms of actuators: across structures the best assignments used three, four and seven actuators at ceilings of one, three and one releases, and on Cross L, where the ceiling was raised from one to three with everything else fixed, the actuator count stayed at four, the bracket ceiling, while the mean placement error fell from 41.55 mm to 11.41 mm (Table 7.6). What a release buys therefore is reach, not a reduction in actuators.

RQ3 asked how the definition of acceptable closeness and the stated design preferences change which assignment is selected and which configuration is reached. Both change it, and each was varied over one frozen candidate population with the other held fixed. The tolerance moved the leading assignment at every step tested and tightening it did not tighten the placement achieved: on Truss W

the leading candidate placed at a mean of 70.91 mm at τ = 0.1 mm against 0.51 mm at τ = 1 mm (Table 7.8). The weight vector moved both the assignment and the actuator count, which fell under pure count weighting from seven to four on Truss W and Short C, from three to two on Panel P and from four to three on Cross L, with mean placement error rising. These end points are the number of target nodes, or the bracket floor on Cross L, because the count subscore gives full marks to every assignment with at most one actuator per target; below that reference the selection is not made by the score.

RQ4 asked whether the assignments and travels prescribed by the framework can be executed on a physical demonstrator without adjustment. Four assignments were built and driven on the two demonstrators, spanning three to seven actuators and including two carrying a released member (Tables 7.11 to 7.14). All four executed: each structure moved under actuation, remained driven throughout the travel and reached a configuration corresponding to the computed one. What the search returns (a role per member and a percentage per actuator) is sufficient to build and drive the structure through the control pipeline with no intermediate adjustment. The physical results establish executability: the realised geometry was not measured, so the agreement between physical and computed configuration is qualitative.

Taken together, these answers state the conditions under which a planar pin-jointed lattice of given topology can be driven, controllably, toward a prescribed target geometry. For lattices at the scale tested (fifteen to nineteen live members), and with each actuator adding an independent degree of freedom, an assignment of roles must satisfy four conditions in sequence:

1. **It must be a driven mechanism.** With its actuators held the frame is rigid, and with them free it has exactly one degree of freedom per actuator. In terms of the frame's states of self-stress (Equation 3.4), every released member must carry self-stress independently of the others, which limits the releases to s_0, and no state of self-stress surviving the releases may load an actuated member. Without releases only members free of self-stress can be actuated; with them, an assignment needs at least one release per independent state of self-stress loading its actuators. On Cross L and Short C this condition could not be met without releases; on Panel P it could, but none of the resulting mechanisms met the second condition.
2. **Its motion must reach the targets.** At least one of the motions the mechanism admits must move the target nodes toward the curve, rather than along it or not at all (Section 5.4.3). A valid mechanism can fail here, as every combination of the four self-stress-free members of Panel P did.
3. **It must have as many independent motions as the target has residuals.** Placing n_t target nodes on a curve with their spacing held sets $2n_t - 1$ residuals, which can generically all be driven to zero only where $n_a \geq 2n_t - 1$ (Section 5.5). The count is necessary rather than sufficient: below it the optimiser returns a least-squares compromise, as on Cross L, and at it a residual can still be left unreachable, as at the end node of Short C.
4. **Its path must be drivable.** From the drawn to the reached configuration, the forward solve must converge, the motion must stay clear of a fold as measured against SING_THRESH, every actuator must stay within its stroke, and every released member must stay within its slider (Section 5.8). No swept candidate in the case set violated this condition, so its thresholds were not tested at their limits.

The first and third conditions can be read before any search is run: the states of self-stress follow from one decomposition of the drawn frame, and the residual count from the number of target nodes. The second and fourth are checked per candidate. Among the assignments satisfying all four, the Rank-1 selected is set by the search preferences: tolerance and scoring weights change the selection and the configuration reached, but not whether adaptation is possible (RQ3). The prescriptions that satisfied them were executed on the demonstrator without adjustment (RQ4). Three limits bound these results. The framework reasons about motion alone, so nothing establishes that a reconfigured structure carries load. The realised geometry of the demonstrator was not measured, so the physical results establish executability, not accuracy. Finally, roles are assigned at assembly rather than switched in the field, so this work is a step toward reconfiguration after deployment rather than a demonstration of it.

9.2. Future development

Four possible future developments are listed in order of how much each guides the framework towards a usable engineering tool.

Statics. Section 8.6 identifies the absence of any structural reasoning as the widest gap between this tool and a usable engineering method. What must be added is member sizing, self-weight, actuator force and at least one load case, evaluated both at the reached configuration and during the actuation rather than only at the drawn one. This would improve the value that a score carries, moving it from a statement of kinematic preference into an engineering criterion, and allowing an assignment that places the targets perfectly and then collapses to be rejected.

A population-based search for combinatorially wide problems. The short-curve case already reached the limit of exhaustive enumeration: of 1.103×10^6 candidates that passed the gates, 576,528 were evaluated within the time budget (Table 7.7). Beyond that point the framework can evaluate only a fraction of the whole admissible set. A genetic or other population method could replace exhaustive enumeration while leaving the gates, the optimiser and the scoring intact, since none of them depends on having seen the whole space.

Extension to three dimensions. The compatibility matrix would gain one column per free node, the free-coordinate count becomes $3(N - |G|)$, and Maxwell's three-dimensional form replaces the planar one. The roles, the two-test, the funnel, the usefulness measure, the scoring and the verification are stated in terms of ranks and residuals, so none requires modification.

Second-order rigidity analysis. A second-order analysis would further resolve the candidates that are mechanisms to first order and rigid at second order, which currently surface only as a failure to converge in the verification sweep.

9.3. Contributions

This thesis makes four contributions.

I. **Conditions for driving a lattice of given topology toward a target.** The four conditions set out above state when an assignment of fixed, actuated and released members drives a lattice to a target: it must form a driven mechanism, move the target nodes toward the curve, carry enough actuators for an exact fit, and follow a drivable path. Release proved decisive: in the three structures with more members than rigidity requires, no assignment met the first two conditions without one, so which members are released decides whether the lattice can be driven toward its target at all. Both results follow from the classical compatibility conditions of the frame; what is contributed is their statement for the inverse problem with three member roles and their test on four planar structures.

II. **A design-automation layer over established first-order kinematic theory.** The rigidity analysis is the matrix formulation of Pellegrino and Calladine (1986), with rank taken by singular value decomposition after Pellegrino (1993). What is contributed is a staged search that constructs the admissible set directly from the actuator and release bounds, applies a rank-based validity test, and discards valid mechanisms that do not move the targets, reducing 3^n candidate role assignments to a verified and ranked set.

III. **A separation of measurement from preference.** The search records a fixed vector of measurements per candidate to a fingerprinted store, and scoring is arithmetic applied to that store afterwards. A change of tolerance, aggregation exponent or weight vector therefore re-ranks a frozen candidate population rather than re-running the search, which is what allowed both sweeps of RQ3 to be read from a single population and attributed to the one parameter each varied.

IV. **A reconfigurable pin-jointed demonstrator and control pipeline** that execute the framework's output on hardware directly: four computed assignments, with three to seven actuators and two carrying a released member, ran without adjustment.

In a static structure redundancy is a structural margin; in an adaptive one it is a budget, which in every redundant frame tested had to be spent before the actuators could move the structure toward its target. Which members to spend it on is now a decision that can be computed rather than taken by hand.

9.4. AI Usage

I used Claude (Anthropic; Opus 5 and 5.5) as an assistant in two areas of this project: the implementation of the computational framework and the review and editing of the manuscript. The research questions, design decisions, structural claims and interpretation of the results are my own. For the framework, Claude assisted with implementing, debugging and refactoring the Python codebase, and with checking technical reasoning against the code. I defined the kinematic formulation, the search and its validity tests, and the evaluation protocol, directed each implementation step, and tested the code before using its output. For the manuscript, Claude reviewed drafts for errors and for consistency between chapters, and suggested clearer wording throughout it. Claude helped implement and articulate the work; it did not originate it.

References

Asimow, L. and B. Roth (1978). “The Rigidity of Graphs”. In: *Transactions of the American Mathematical Society* 245, pp. 279–289. DOI: 10.2307/1998867.

Bironт, Andreas (2022). “Design of Cellular Structures for Robotic Assembly”. MSc thesis. Delft University of Technology. URL: https://repository.tudelft.nl/record/uuid:34af651c-72de-4e99-aa9c-a97e7c0be0ce (visited on 12/07/2025).

Bricard, Raoul (1897). “Mémoire sur la théorie de l’octaèdre articulé”. French. In: *Journal de Mathématiques Pures et Appliquées*. 5th ser. 3, pp. 113–148. URL: https://www.numdam.org/item/JMPA_1897_5_3__113_0/.

Calladine, C. R. (1978). “Buckminster Fuller’s ‘Tensegrity’ structures and Clerk Maxwell’s rules for the construction of stiff frames”. In: *International Journal of Solids and Structures* 14.2, pp. 161–172. DOI: 10.1016/0020-7683(78)90052-5.

Calladine, C. R. and S. Pellegrino (1991). “First-order infinitesimal mechanisms”. In: *International Journal of Solids and Structures* 27.4, pp. 505–515. DOI: 10.1016/0020-7683(91)90137-5.

Cramer, Nicholas B. et al. (2019). “Elastic shape morphing of ultralight structures by programmable assembly”. In: *Smart Materials and Structures* 28.5, p. 055006. DOI: 10.1088/1361-665X/ab0ea2.

Feng, Li-Jia, Lin-Zhi Wu, and Guo-Cai Yu (2016). “An Hourglass truss lattice structure and its mechanical performances”. In: *Materials & Design* 99, pp. 581–591. DOI: 10.1016/j.matdes.2016.03.100.

Gawde, Pranav Ravindra (2021). “Modularity in Lattice Structures for Circular Product Design”. MSc thesis. Delft University of Technology. URL: https://repository.tudelft.nl/record/uuid:b0fc5f9a-b4e1-455a-8bef-1c9a48054d87 (visited on 12/07/2025).

Gorguluarslan, Recep M. et al. (2016). “Design and fabrication of periodic lattice-based cellular structures”. In: *Computer-Aided Design and Applications* 13.1, pp. 50–62. DOI: 10.1080/16864360.2015.1059194.

Helou, Mark and Sami Kara (2018). “Design, analysis and manufacturing of lattice structures: an overview”. In: *International Journal of Computer Integrated Manufacturing* 31.3, pp. 243–261. DOI: 10.1080/0951192X.2017.1407456.

Hillrom (2026). *Compella Bariatric Bed*. URL: https://www.hillrom.com.au/en/products/compella-bariatric-bed/ (visited on 09/23/2026).

Jenett, Benjamin et al. (2016). “Meso-Scale Digital Materials: Modular, Reconfigurable, Lattice-Based Structures”. In: *Volume 2: Materials; Biomanufacturing; Properties, Applications and Systems; Sustainable Manufacturing*. American Society of Mechanical Engineers. Blacksburg, Virginia, USA, V002T01A018. DOI: 10.1115/MSEC2016-8767.

Jenett, Benjamin et al. (2017). “Digital Morphing Wing: Active Wing Shaping Concept Using Composite Lattice-Based Cellular Structures”. In: *Soft Robotics* 4.1, pp. 33–48. DOI: 10.1089/soro.2016.0032.

Jenett, Benjamin Eric (2020). “Discrete Mechanical Metamaterials”. PhD thesis. Massachusetts Institute of Technology. URL: https://dspace.mit.edu/handle/1721.1/130610 (visited on 10/12/2026).

Kim, Jiheon (2020). “Repairability of lattice structure in a circular economy”. MSc thesis. Delft University of Technology. URL: https://repository.tudelft.nl/record/uuid:77a6bfb9-3388-4688-aee9-3dfd10994dc3 (visited on 12/07/2025).

Kumar, P. and S. Pellegrino (2000). “Computation of kinematic paths and bifurcation points”. In: *International Journal of Solids and Structures* 37.46, pp. 7003–7027. DOI: 10.1016/S0020-7683(99)00327-3.

Levenberg, Kenneth (1944). “A method for the solution of certain non-linear problems in least squares”. In: *Quarterly of Applied Mathematics* 2.2, pp. 164–168. DOI: 10.1090/qam/10666.

Marquardt, Donald W. (1963). “An Algorithm for Least-Squares Estimation of Nonlinear Parameters”. In: *Journal of the Society for Industrial and Applied Mathematics* 11.2, pp. 431–441. DOI: 10.1137/0111030.

Maxwell, J. Clerk (1864). “On the calculation of the equilibrium and stiffness of frames”. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 27.182, pp. 294–299. DOI: 10.1080/14786446408643668.

Miura, Koryo and Hiroshi Furuya (1988). “Adaptive structure concept for future space applications”. In: *AIAA Journal* 26.8, pp. 995–1002. DOI: 10.2514/3.10002.

Nelder, J. A. and R. Mead (1965). “A Simplex Method for Function Minimization”. In: *The Computer Journal* 7.4, pp. 308–313. DOI: 10.1093/comjnl/7.4.308.

Pellegrino, S. (1993). “Structural computations with the singular value decomposition of the equilibrium matrix”. In: *International Journal of Solids and Structures* 30.21, pp. 3025–3035. DOI: 10.1016/0020-7683(93)90210-X.

Pellegrino, S. and C. R. Calladine (1986). “Matrix analysis of statically and kinematically indeterminate frameworks”. In: *International Journal of Solids and Structures* 22.4, pp. 409–428. DOI: 10.1016/0020-7683(86)90014-4.

Sachse, Renate et al. (2021). “Motion Design with Efficient Actuator Placement for Adaptive Structures that Perform Large Deformations”. In: *Frontiers in Built Environment* 7, p. 545962. DOI: 10.3389/fbuil.2021.545962.

Senatore, Gennaro, Philippe Duffour, and Peter Winslow (2019). “Synthesis of minimum energy adaptive structures”. In: *Structural and Multidisciplinary Optimization* 60.3, pp. 849–877. DOI: 10.1007/s00158-019-02224-8.

Shen, Zhangming, Difeng Zhu, and Mingchao Zhang (2025). “Synergy of Smart Materials and Structures Toward Intelligent Metamaterials”. In: *SmartSys* 1.3, e70007. DOI: 10.1002/sys3.70007.

Sofla, A. Y. N., D. M. Elzey, and H. N. G. Wadley (2009). “Shape morphing hinged truss structures”. In: *Smart Materials and Structures* 18.6, p. 065012. DOI: 10.1088/0964-1726/18/6/065012.

Wang, Zhigang et al. (2022). “Design of a Distributedly Active Morphing Wing Based on Digital Metamaterials”. In: *Aerospace* 9.12, p. 762. DOI: 10.3390/aerospace9120762.

Zhao, Luyang et al. (2023). “StarBlocks: Soft Actuated Self-Connecting Blocks for Building Deformable Lattice Structures”. In: *IEEE Robotics and Automation Letters* 8.8, pp. 4521–4528. DOI: 10.1109/LRA.2023.3284361.

Zhao, Luyang et al. (2025). “Modular shape-changing tensegrity-blocks enable self-assembling robotic structures”. In: *Nature Communications* 16, p. 5888. DOI: 10.1038/s41467-025-60982-0.

A

Design requirements

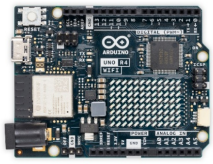
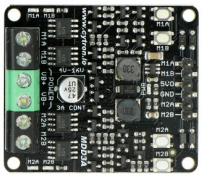
Table A.1: Design requirements of the assembly, the nodes and the beams.

Component	N°	Requirement	Related
Structure	A1	The assembly needs to perform planar transformations	*B1, N1, N3, N4, N6*
	A2	The assembly can be transformed with the distribution of its own weight on a supporting plane/area	*B3, N7*
	A3	The assembly needs to be able to support/actuate its own weight	*B3, N7*
	A4	The assembly must allow a simple disassembly of its elements	*N2, N5*
	A5	Putting the assembly together must be simple	*B2, N5*
	A6	The assembly's movement must be controlled and intentional	*B4*
	A7	The assembly must perform different transformations by the combination of the single elements' transformations	*B4, N1, N4, N6*
Nodes	N1	The nodes must allow movement in a 2D environment	
	N2	The nodes must allow beams to be inserted successfully	
	N3	The nodes must allow only one rotational degree of freedom	
	N4	The nodes must move freely across their full range of motion, without binding or locking	
	N5	The nodes can work with different beam interfaces	
	N6	The rotation axis between two connected beams must be normal to the plane of the assembly	
	N7	The nodes must support the beams connected	
Beams	B1	The beams, when passive, must allow contractions and elongations	
	B2	The beams must fit within the interface provided by the nodes	
	B3	The beams must be strong enough to support the forces imposed by the actuations across the structure	
	B4	The actuated beams must be controllable	

B

Demonstrator components

Table B.1: Electronic components of the demonstrator.

Component	Datasheet	Notes
Arduino UNO R4 WiFi 	• Operating voltage: 5 V • Flash memory: 256 kB • SRAM: 32 kB • Data memory (EEPROM): 8 kB • Power: USB-C at 5 V • Digital I/O pins: 14 • Analog input pins: 6 • Length: 68.85 mm • Width: 53.34 mm	Although the board provides a Wi-Fi module that would allow wireless communication with the PC, this would have added a further layer of complexity and was therefore not used. Each actuator requires two outputs, so a maximum of eight can be driven from a single board, using the analog pins as digital outputs alongside the digital ones.
4× Cytron MDD3A 	• Operating voltage: 4–16 V • Continuous current: 3 A per channel • Peak current: 5 A per channel, for short durations • Channels: 2 • On-board test button for each channel • Stackable with 30 mm clearance	This component was chosen for its dual-channel layout, which reduces the total number of driver boards required, and for its current headroom relative to the actuators used in the project.
Linear actuator 	• Input voltage: 12 V • Stroke force: 70 N • Stroke length: 100 mm • Operating temperature: 0–40 °C • Limit switches: internal • Working speed: 8 mm/s • Guide rail type: T-screw • IP rating: IP60	`https://www.amazon.nl/dp/B0B8JTNJWK`
16× JST SM 2-pin connector 	• Cable length: 150 mm • Wire gauge: 22 AWG	`https://www.amazon.nl/dp/B0CBWXG6JT`

C

The singularity threshold

What it represents

While a structure moves, it can approach a fold. A fold is a position, such as two bars lining up, in which the actuators no longer fully decide where a joint goes. At every step of the motion, the solver measures how far the structure is from such a position with a single number, κ, between 0 and 1. A value close to 1 means the actuators are firmly in control. A value close to 0 means the structure is near a fold.

Near a fold, small imperfections are magnified (Figure C.1). A slightly inaccurate actuator or a loose pin turns into a large error in where the joints end up, and that error is different every time the motion is repeated. The singularity threshold, SING_THRESH, is the lowest value of κ the built structure can accept before this error exceeds what the application tolerates. Any motion that drops below it is rejected. κ also depends on the lengths of the members, so the same value means different things on different structures, and the threshold is worked out for each structure separately.

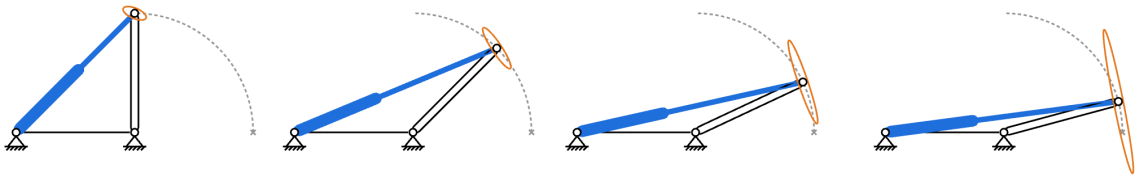


Figure C.1: A two-bar linkage driven towards a fold. The actuator (blue) extends and the free joint moves along the dashed circle about the right support, towards the position where all three joints line up (cross). The ellipse shows where the free joint can end up for length errors of the same size in both members. It grows as the fold approaches and stretches in the direction neither member resists. The ellipses are enlarged, but their sizes relative to each other are to scale.

How to obtain it

The threshold follows from three things: how precise the parts are, how large the structure is, and how precise the result must be:

$$\kappa = \frac{\sqrt{\sum_k \left(L_k\,\delta_k\right)^2}}{\tau \cdot \max_k \left(L_k \sqrt{n_k}\right)} \tag{C.1}$$

The symbols mean the following:

- τ is the placement tolerance in millimetres: how far a target joint may land from the curve. It is the same value used in scoring.
- L_k is the length of member k.
- δ_k is the largest length error member k can have. For an actuator, this is how far it can land from its commanded length plus the play in its two pins. For a fixed member, it is the play in its two

pins only.

- n_k is the number of ends of member k that are not on a support, so either 1 or 2.

The sum and the maximum run only over members that hold their length, meaning fixed members and actuators. Released members and members joining two supports are left out.

The top of the fraction is the total uncertainty the members bring, with longer members counting more. The bottom is the tolerance, scaled by the size of the largest member.

The value is obtained in four steps:

1. Choose τ from what the application needs.
2. Find each δ_k. Measure how far each actuator lands from its commanded length over repeated strokes, and measure the play in the pins.
3. Compute ρ_{th} using all members of the structure. Enter it in the SING_THRESH input and re-rank the stored designs (Section 5.7). No new search is needed.
4. Once a design has been chosen, repeat the calculation using only its members that are not released. Do it once with the actuator lengths at the start of the motion and once with their lengths at the end, and keep the larger result. Confirm that the design's lowest ρ during the motion stays above this value.

If κ comes out above 1, no position can meet the tolerance with these parts. Either accept a larger τ, or use more precise actuators and tighter pins.

The threshold also affects scoring. The path score reaches its maximum only for designs whose lowest κ is at least one hundred times the threshold. With a strict threshold, no design reaches that maximum, so this part of the score counts for less than its weight suggests.

Checking it on a physical rig

The formula assumes the parts behave as specified, but friction and backlash under load can make them worse. To check this, a two-bar linkage should be built from the same actuators and pins, then repeatedly driven to positions closer and closer to its fold, where the two bars line up. Measure, at each position, how much the end position varies between repetitions.

The κ at which this variation reaches τ is the real threshold of the hardware. If it is higher than Equation (C.1) predicts, increase the δ_k values until the equation reproduces it. Then use those values for every structure built from the same parts.