Department of Precision and Microsystems Engineering

The Development and Geometric Analysis of an Origami-based Constant-height Walking Locomotion System

Jim Sluijter

Report no	: 2023.025
Coach	: Dr. ir. D. (Davood) Farhadi
Professor	: Dr. ir. D. (Davood) Farhadi
Specialisation	: Mechatronic System Design (MSD)
Type of report	: Thesis report
Date	: April 13 2023

Delft
University of
Technology

Challenge the future

The Development and Geometric Analysis of an Origami-based Constant-height Walking Locomotion System

by

J. Sluijter

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended publicly on Thursday April 13th at 15:00

Student number:	4286898	
Project duration:	September, 2021 – April, 2023	
Thesis committee:	Dr. ir. D. (Davood) Farhadi	TU Delft, supervisor
	Prof. dr. ir. J.L. (Just) Herder,	TU Delft, chair
	Dr. ir. A.M. (Alejandro) Aragón,	TU Delft

TU Delft

Preface

This thesis not only concludes my Masters degree in High-Tech engineering, but also marks the end of long journey of studying at the Delft University of Technology.

First of all, I would like to express my gratitude towards Davood Farhadi for introducing me to the topic of origami-based engineering, for providing me feedback and challenging my ideas, and for providing much needed guidance and motivation throughout my project. I would also like to thank Just Herder, for going out of his way to chair my graduation, and for providing me valuable feedback on my report. Furthermore, I would like to thank everyone who provided me with support during the project.

Contents

Chapter 1

Technical paper

The Development and Geometric Analysis of an Origami-based Constant-height Walking Locomotion System

J. Sluijter

March 29, 2023

Abstract

Origami-based mechanisms have emerged as a promising solution for developing locomotion systems. Its light-weight nature, scalability, and possibility for 2D monolithic manufacturing makes origami an attractive option for various applications, such as mobile robots and meta-surfaces. While various types of origami-based locomotion exist, constant-height walking locomotion does not have an origami-based solution yet. To achieve this, we present a 2-DOF crease pattern with two internal vertices and geometric constraints that can perform the required output path. A parametric study performed on the presented pattern reveals many different feasible geometries. Subsequently, these geometries are evaluated based on their capabilities to produce a path with minimized displacement along the Z-direction and minimal change in velocity of the end-effector during propulsion. As a demonstration, we then utilized these results to optimize the design of a locomotion system for active surfaces. Finally, the results are verified with experiments using a physical prototype. In conclusion, the analysis of the results provides valuable insights in the behavior of the crease pattern, which can be utilized for designing and optimizing an origami-based locomotion system for other applications with different requirements.

1 Introduction

Origami-based mechanisms have a great potential for application in locomotion systems with various purposes. For instance, origami is a viable option for use in meso-scale mobile terrestrial robots, as its advantages can complement the advantageous properties of these robots. Due to their thin-paneled structure, origami-based robots can be made light-weight, making them energy efficient, cheap to transport, and resistant to falling damage. Furthermore, origami-based robots can be folded flat for compact storage during transportation, and independently deployed when needed [1, 2]. This makes origami-based robots an ideal candidate for swarm robotics in space applications [3], where large quantities of robots can be compactly stacked during transportation.

Besides application in mobile robots, origami-based locomotion could also be inverted to be utilized in the design of an *active surface*. When a crease pattern suitable for locomotion is tessellated on this surface, it can be utilized to manipulate an object on top of it. As origami can be manufactured monolithically from a single sheet of material, with low-cost 2D manufacturing techniques, integrated actuation, and integrated sensing [4], a large tessellated pattern with many complex features can be manufactured in a cost-effective way without the need for assembly. Additionally, the possibility for 2D manufacturing allows for the use of high-precision manufacturing techniques such as laser cutting, water cutting, wire EDM, or photolithography [5–8], which enable to scaling the mechanisms down to micro-scale dimensions

Current origami-inspired locomotion systems use a variety of locomotion principles, which can be categorized into crawling, walking, hopping, and short-stride vibration-based propulsion. The majority of these systems use origami to perform a crawling locomotion [9–17]. However, crawling locomotion and vibration-based propulsion [18, 19] suffer inherently from slipping over the contact surface, leading to reduced energy efficiency and limiting their maximum velocity. Moreover, when applied to mobile robots, crawling motion offers limited maneuverability, and renders these robots incapable of navigating uneven terrain with obstacles and sudden changes in height. While hopping locomotion [9, 20] could potentially overcome the issue of navigating uneven terrain, it is difficult to control and offers limited stability.

Walking locomotion, on the other hand, does not have these limitations. While origami-inspired mechanisms exist that utilize a form of walking, none of the existing designs are truly origami-based, i.e., folded from a single sheet of paper without cuts. They fail to meet the essential criteria of true origami, for instance because they are comprised of multiple laminae [21, 22], or because their topology can only be created by cutting the material [23, 24]. Other designs require non-origami-based mechanical parts to function cite [25].

Moreover, no origami-based topology exists that can perform constant-height walking. This form of walking, where the height between the base and end effector of the mechanism does not change during the propulsion phase, is particularly promising. Because the object or robot does not change height, a stable, energy efficient movement is achieved.

Therefore, in this paper we design a truly origami-based mechanism that exhibits constant-height walking locomotion. In order to achieve this, a 2-DOF crease pattern topology with two internal vertices is proposed. Subsequently, following a study on the workspace of the mechanism, this topology is subjected to a set of geometrical constraints, resulting in a crease pattern with guaranteed possibility for a straight-line output. This crease pattern is then subjected to a parametric study, which is used to find all geometries that generate a feasible output path for a prescribed coupled input. These resulting feasible paths are evaluated based on criteria, in order to generate knowledge on the behavior of the output path for different geometries. For demonstration, this knowledge is used to select an optimal geometry for a given design case: an active surface for high-precision manipulation. Finally, the results are validated through experimentation on a physical prototype.

This paper is structured as follows. In Section 2, the results of the topology synthesis (2.1) and the method of finding the suitable crease pattern (2.2) are described, as well as the procedure of the parametric geometric study (2.3). Section 3 describes and discusses the results of the overall qualitative behavior (3.1) of this crease pattern, as well as the local quantitative behavior of the selected class, which is used to select a final design for the proposed design case in Section 3.2. The paper is concluded with an overall discussion of the findings in Section 4, and a summary of the most important findings in Section 5. Detailed descriptions of the methodology regarding the concept generation, topology synthesis, parametric study algorithm, and prototyping and evaluation can be found in the supplementary materials.

2 Method

Firstly, a crease pattern is developed that is able to mimic walking locomotion. Subsequently, this crease pattern is adjusted such that it can facilitate constant-height walking.

In order to perform walking locomotion, the mechanism needs to exert a cyclical output path with a *propulsion phase*, and a *return phase*. During the propulsion phase, the end-effector mechanism is in contact with the surface, and exerts an in-plane force on this surface such that it propels itself. The closed loop of the output path is completed during the return phase, such that the end effector can return to the start of the propulsion phase without colliding with the locomotion surface. To facilitate the use of rigid-facet origami, and to avoid passing through a possible singularity point, the cyclical path is achieved by using a crease pattern with two kinematic degrees of freedom.

Furthermore, to achieve constant-height walking, the propulsion phase of the output path should consist of a straight horizontal line. This can be accomplished through proper control of the two independent input signals, or by optimizing the geometry of the crease pattern such that it follows the desired path for a prescribed input. An active surface often exists of many different units performing identical tasks. Therefore, in order to minimize the amount of inputs, a coupled input with an optimized geometry is preferred. On the contrary, independent control of the input allows greater versatility for origami-based mobile robots that are required to perform complex tasks or navigate uneven terrain with obstacles or height differences.

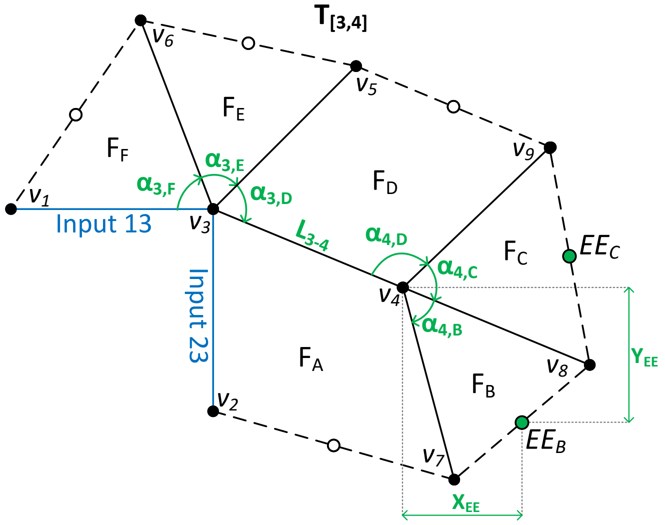


Figure 1: One of the two 2-DOF topologies selected from the results: $T_{[3,4]}$. End effector candidates are highlighted in green. The other result ($T_{[3,5]}$), can be found in Appendix B.

2.1 2-DOF crease pattern topology

To create a fitting 2-DOF crease pattern topology that can exert a cyclical path, a computational synthesis method by Zimmerman et al. [26] has been reproduced. This method uses the Principle of Three Units (PTU), combined with graph theory, to automatically generate crease pattern graphs by extending and merging nodes. To fit the objective of this paper, the algorithm is adjusted at several instances. A full description of these adjustments is provided in Appendix B.

Multiple feasible crease pattern topologies emerge from the application of the algorithm. After evaluation, topology $T_{[3,4]}$ (Figure 1), generated by expanding nodes v_3 and v_4, is selected for further development. This topology has two auxiliary vertices, located on facets F_B and F_C, that can function as end effector. Since the behavior of the end effectors is only affected by their predecessors, both end effector candidates in the crease pattern can be described using 9 parameters:

1. the sector angles of the first, fixed internal vertex ($\alpha_{3,F}$, $\alpha_{3,E}$, and $\alpha_{3,D}$)
2. the sector angles of the second, floating internal vertex ($\alpha_{4,D}$, $\alpha_{4,C}$, and $\alpha_{4,B}$)
3. the length of the crease-line between these two internal vertices (L_{3-4})
4. the coordinates of the end-effector relative to the floating internal vertex (X_{EE} and Y_{EE})

The exact position of the remaining boundary vertices does not affect the behavior of the mechanism, and can therefore be freely chosen as long as the mentioned parameters are taken into account. An overview of the other resulting topologies is provided in Appendix B.

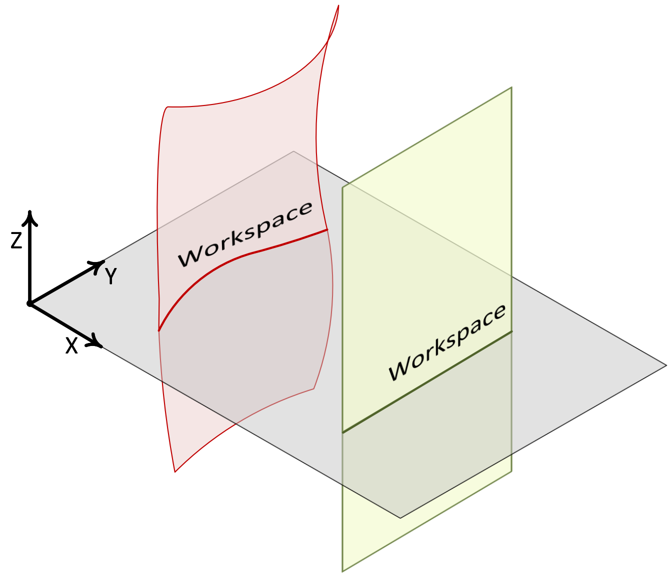


Figure 2: Intersection of the workspace with a horizontal plane. Left: an arbitrary workspace with no straight-line intersection. Right: a workspace that lies entirely in a plane orthogonal to the horizontal plane, resulting in a straight-line intersection at any height inside the workspace

.

2.2 Crease pattern with straight-line output

To find a suitable geometry for the selected topology, such that a straight-line propulsion section can be achieved, a preliminary study has been conducted in which the workspace of the end effector candidates is examined.

For the chosen 2-DOF origami-based mechanisms, the workspace of any end effector lies on a 2D-surface in three dimensional space. At the intersection of the workspace with a horizontal plane, we find the horizontal trajectory that the mechanism is able to exert at the Z-coordinate of this plane (Figure 2). Therefore, it can be concluded that a mechanism is able to exert the desired horizontal straight-line motion if, and only if, there exist a Z-coordinate for which the intersection between its workspace and the horizontal plane is a straight line section. Consequently, when the workspace lies entirely in a single plane, the corresponding mechanism will be able to exert the required motion at any height within its workspace. Such a planar workspace can be achieved by subjecting chosen topology $T_{[3,4]}$ to geometric constraints. When the orientation of crease c_{4-7} is constrained to be parallel to input crease c_{2-3}, the workspace of any point on facet F_B (represented by EE_B) will lie entirely on a plane parallel to the XZ-plane.

Applying this constraint results in the crease pattern presented in Figure 3a. The implemented constraint simplifies the model by restricting it to a single end effector candidate EE_B, and by eliminating two parameters. Firstly, the constraint sets sector angle $\alpha_{4,A}$ to be dependent on $\alpha_{3,A}$, which is in turn dependent on the variable sector angles around v_3). Moreover, since any point on facet F_B is restricted to movement in the XZ-plane the Y-coordinate Y_{EE} of the end effector does no longer affect its behavior and can be discarded as a parameters.

The resulting mechanism can be considered equivalent to a planar linkage mechanism with RR topology moving in the XZ-plane (Fig. 3b). This can be used to simplify the calculation of the position $\mathbf{P_{EE}}$ of the end effector, greatly reducing the computational costs to perform the parametric study. In this equivalent mechanism, rigid links R_A and R_B represent facets F_A and F_B respectively, while the crease lines between them are represented by the two revolute joints. This allows for a great simplification of the calculation of $\mathbf{P_{EE}}$, from a large system of three dimensional matrix equations, to a single trigonometric expression that the 2D position $\tilde{\mathbf{P}}_{\mathbf{EE}}$ on the XY-plane:

$$\tilde{\mathbf{P}}_{\mathbf{EE}} = \begin{bmatrix} X_{EE} \\ Z_{EE} \end{bmatrix} = R_A \begin{bmatrix} \cos(\rho_{2-3}) \\ \sin(\rho_{2-3}) \end{bmatrix} + R_B \begin{bmatrix} \cos(\rho_{2-3} + \rho_{4-7}) \\ \sin(\rho_{2-3} + \rho_{4-7}) \end{bmatrix} \tag{1}$$

where fold angle ρ_{2-3} is a direct input, and ρ_{4-7} is a function of both inputs ρ_{1-3} and ρ_{2-3}, calculated with the PTU method.

Because all points on facet F_B with the same distance R_B to crease c_{4-7} exert an identical motion, the end effector can be represented as a line instead of a single point, which facilitates line contact during propulsion. Besides providing better mechanical properties by distributing the load, this also decreases the number of legs needed for a mobile robot by offering multiple contact points per leg. Since stable walking requires at least three contact points to the ground, utilizing line contact decreases the amount of legs that need contact with the ground at any given instance from three to two, provided the legs are properly positioned around the center of mass of the robot.

Furthermore, since we have not yet defined the dimensions of the crease pattern, a third parameter can be removed by normalizing the dimension of the facet F_A to $R_A = 1$. This eliminates the need to define L_{3-4}, leaving R_B as the only dimensional parameter.

For a given configuration of the input folds, every vertex of the crease pattern has two rigidly foldable configurations (denoted as *True* or *False*), in which the fold angles can be orientated, resulting in four possible Rigid Body Modes (RBMs): RBM_{TT}, RBM_{TF}, RBM_{FT}, and RBM_{FF}. An example of the four RBMs for a specific geometry can be seen in Figure 3c. However, this paper classifies the modes according to the vertex triangle direction as used in the Principle of Three Units [27]. Therefore, the behavior of mechanisms assigned to the same RBM can exhibit different behavior dependent on their geometry.

2.3 Parametric study

While the proposed crease pattern with independent control of the two degrees of freedom in the system could be sufficient for a mobile robot, it is favorable for an active surface to minimize the amount of independent input signals because it exists of many tessellated units. Therefore, a parametric study is performed, which is used to find an optimized geometry for a given application of an active surface. In this study, the six free geometric parameters Ω in the crease pattern are varied for every of the four available Rigid Body Modes. The five sector angles are constrained to lie between $10°$ and and $170°$ in order to avoid facets that are too narrow or too wide, while the dimensional parameter is varied between 0.6 and 3.0. The step sizes for these parameters ($\Delta\alpha$ & ΔR) are based on the available computational resources. This results in the following design space:

$$\begin{aligned}\Omega_\alpha &= \{\alpha_{3,F}, \alpha_{3,E}, \alpha_{3,D}, \alpha_{4,D}, \alpha_{4,C}\} \in [10° - 170°]\\ \Omega_R &= \{R_A\} \in [0.6 - 3.0]\\ \Omega &= \Omega_\alpha \cup \Omega_R\end{aligned}$$

For every of the four available RBMs, all points in the design space are assessed to determine if they can produce a physically valid, rigidly foldable configuration for the entire given input range. From resulting computed paths, all feasible paths that satisfy the requirements to produce a constant-height walking motion are selected. The paths are then evaluated based on criteria, in order to acquire optimal performance for the specified application: an active surface for precision manipulation.

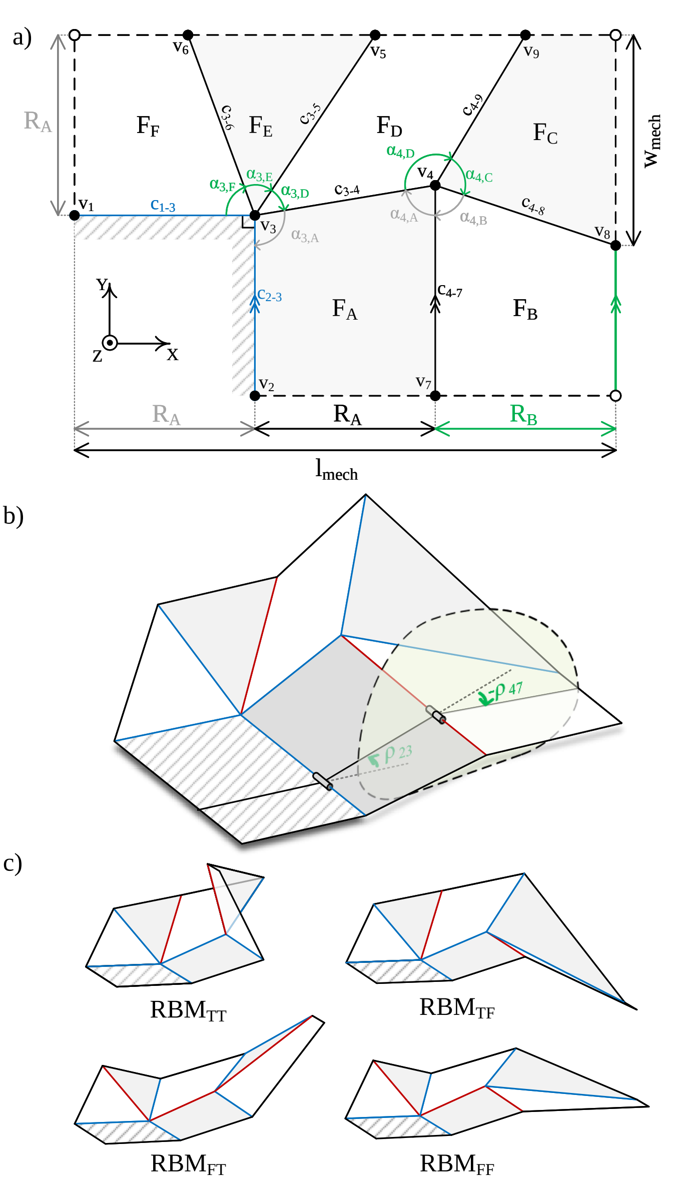


Figure 3: a) The resulting crease pattern with workspace in the XZ-plane. b) The equivalence of the crease pattern with a planar linkage system with in-series RR topology. ρ_{23} has a direct input, while the angle ρ_{47} is determined by the transmission of inputs ρ_{13} and ρ_{23} through vertices v_3 and v_4. c) An example of the four available Rigid Body Modes of a crease pattern

.

Input

The parametric sweep is conducted for a coupled input profile, which consists of two sinusoidal functions ranging from $15°$ to $80°$. The input angle for crease c_{2_3}, has a phase shift of $\phi = +\frac{\pi}{2}$ compared to the input for c_{1_3}.

The phase shift is selected such that the mechanism can perform in an active surface formed by the tessellated four-leg unit cells, as shown in (Fig. 4). In a unit cell, the constituent legs operate with a phase shift $\phi = \frac{\pi}{2}$ with respect to each other, such that they alternately propel the object. By aligning these phase shifts, a single four-phase sinusoidal input can be used to control all units manipulating the object in a single direction. When the direction of the phase between the two inputs for a given leg switches, the end effector direction shifts, while maintaining the output shape.

To maximize the number of feasible paths with a sufficient stride length, the input angles are selected within a lower and upper bound. The lower bound of $15°$ is determined in order to prevent actuated node v_3 from folding flat and entering a singularity. Additionally, the upper bound of $80°$ exists to prevent non-rigidly foldable or intersecting geometries. If the algorithm identifies any position as invalid at any point during the motion's duration, the entire output path is discarded. Both the risk of non-rigidly foldable configurations and the risk of self-intersections increase with a higher fold angle.

Physical validity

For a geometry to be considered physically valid, the sector angles must result in a valid geometry, and the mechanism should be rigidly foldable for over the entire path without self-intersecting.

To maintain geometric validity, the sector angles around a single vertex should not exceed $360°$. Moreover, the remaining dependent sector angles must have a minimum value of $10°$, resulting in the following conditions:

$$\alpha_{3,F} + \alpha_{3,E} + \alpha_{3,D} \leq 260° \tag{2}$$

$$\alpha_{3,F} + \alpha_{3,E} + \alpha_{3,D} + \alpha_{4,D} + \alpha_{4,C} \leq 440° \tag{3}$$

As described by Zimmermann [26] the rigid foldability can be checked during the PTU calculations, by examining if a vertex triangle can exist, using the triangle inequality:

$$U_{max} \leq U_{med} + U_{min} \tag{4}$$

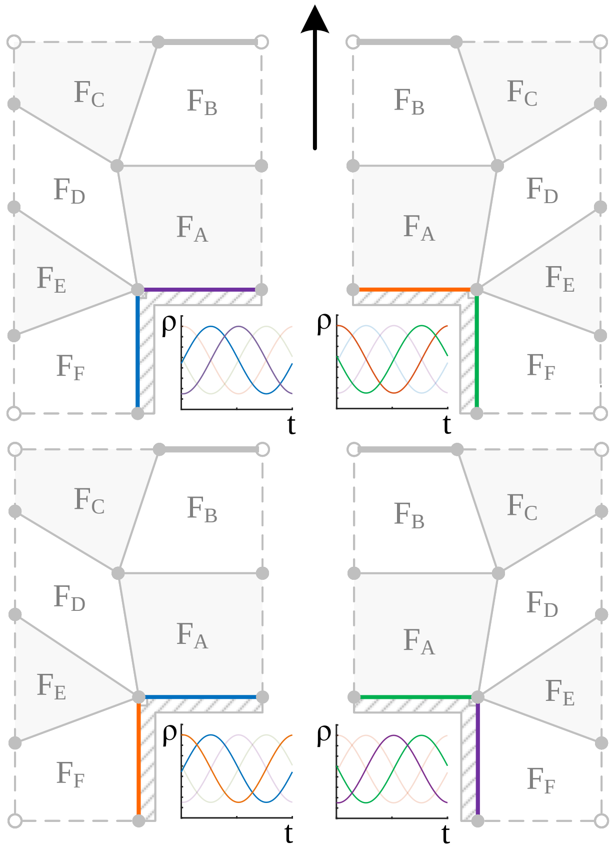


Figure 4: A single four-leg unit cell of the active surface, in the state where it moves an object upwards. For each leg, the two components of the signal used to control fold angle ρ of its input creases are highlighted. Corresponding colors between creases indicate a corresponding phase shift.

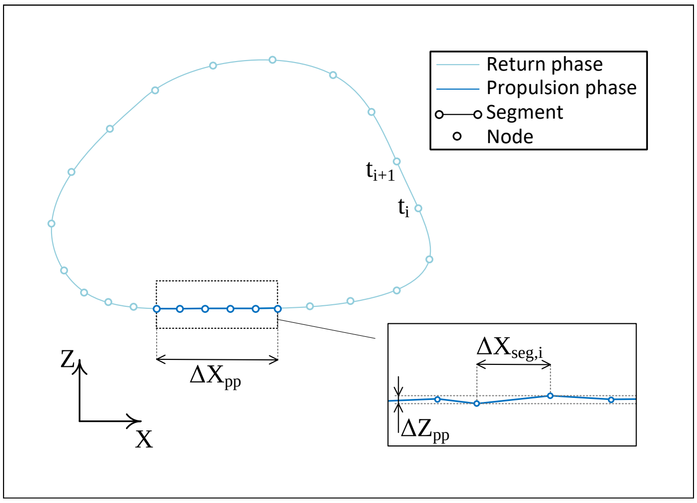


Figure 5: A gait cycle with horizontal section, with accompanying dimensions used in the criteria definitions.

Two types of self-intersection can occur. Firstly, intersection between neighbouring facets can be avoided by limiting the fold angles of all crease lines to a range of $[-180°180°]$. To account for facet thickness this range is decreased to $[-170°170°]$. Furthermore, intersections between non-neighbouring facets is avoided by discarding mechanisms where crease line c_{3-5} intersects with the base plate. While more sophisticated detection algorithms exist, this method discards the vast majority of intersecting mechanisms, while maintaining optimal computational performance.

Path feasibility

The resulting paths generated by the physically valid geometries are examined to determine their feasibility to function as a constant-height walking mechanism. Several conditions must be met for feasibility.

Firstly, a straight horizontal section larger than minimum length C_{min} should be present in the output path. Since the mechanism's dimensions and orientation are not yet determined, the location of this section is not considered in the evaluation. This means the straight section, and with it the locomotion plane, can be either above or below the fixed facet. However, to avoid collisions of the movable object with the fixed facet, while taking into account imperfections in the physical implementation, a minimum absolute Z-coordinate $|Z_{LP}|$ is maintained.

Furthermore, the straight section of the paths should be either the highest or lowest section of the path, in order to avoid the end effector lifting the object. Likewise, no other parts of the mechanism can collide with the locomotion plane.

Lastly, in order to avoid a flat-folded vertex in singularity, and the uncontrollable switching between Rigid Body Modes, a constraint is applied such that all geometries are discarded where all crease lines around v_4 simultaneously reach a fold angle lower than 10 degrees.

Grading criteria

In order to evaluate the performance of the remaining feasible paths, a set of criteria is defined. This is then used to choose an optimal solution for the given design case. A description of dimensions used to define these criteria is shown in Figure 5.

Contact ratio – To define the minimum amount of legs needed to manipulate the object, the contact ratio is defined. This criterion describes how long the leg is in contact with the object, compared to its total path. For a finite-resolution calculation of the paths coordinates, it is determined by the ratio of horizontal segments compared to the total number of generated segments:

$$C = \frac{T_{pp}}{T_{tot}} = \frac{N_{s,pp}}{N_{s,tot}} \tag{5}$$

Vertical error – For an active surface that manipulates the object accurately in-plane, and to prevent the object to move in unwanted directions, the displacement in Z-direction during the propulsion phase should be minimized. Therefore, error criterion ϵ_z defines the error in vertical displacement, normalized with the length of the propulsion phase.

$$\epsilon_z = \frac{\Delta Z_{pp}}{\Delta X_{pp}} \tag{6}$$

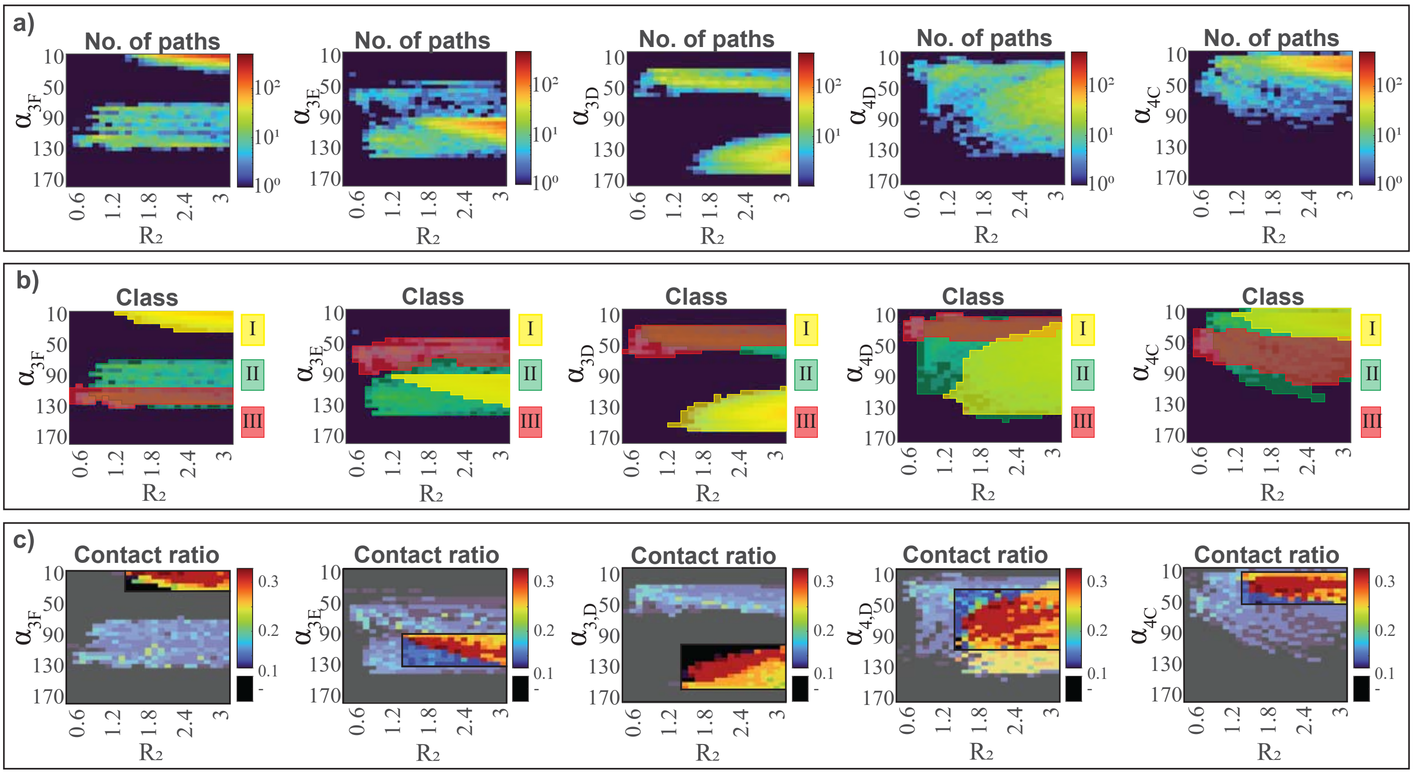


Figure 6: Collection of heat maps displaying the criteria scores in the six-dimensional design space of Rigid Body Mode RBM_{FT}. Each figure displays the optimal characteristics found in the 2D design space of a combination of one of the sector angles with R2. A full overview of all criteria scores can be found in Appendix E. a) Number of feasible paths, b) The sub-regions of the design space where the three geometric classes exist, c) The maximum value of contact ratio C. The region of interest based on this contact ratio is highlighted.

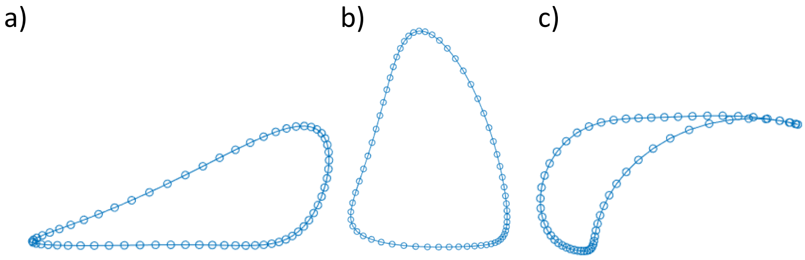


Figure 7: The three output paths resulting from a) Class I, b) Class II, c) Class III

Velocity error – Furthermore, in order to avoid inaccuracies due to hysteresis in manipulation by the active surface, slipping should be minimized. To achieve this, given that multiple legs work cooperate out-of-phase to propel the object, it is important to maintain constant velocity of the end effector in X-direction v_x during the propulsion phase. Therefore, error criterion ϵ_v, which defines the normalized maximum difference in during the propulsion phase, should be minimized.

$$\epsilon_v = \frac{\max(\Delta X_{seg}) - \min(\Delta X_{seg})}{\text{mean}(\Delta X_{seg})} \tag{7}$$

Stride length – Lastly, the stride length of the leg is defined by the length of the propulsion phase, normalized by the size of the mechanism. As the size of the mechanism is not yet determined, and will depend on secondary requirements, this normalization is performed with the minimum size necessary for the mechanism to function. The length of the mechanism l_{mech} depends solely on length R_B, while the width is determined by the minimum Y-coordinate of the end-effector. Both sizes are complemented with a standardized length minimum length, in order to guarantee sufficient functioning of the mechanism (as shown in Figure 3a):

$$l_{mech} = 2R_A + R_B \tag{8}$$

$$w_{mech} = R_A + \left| \frac{R_A}{tan(\alpha_{3,A})} + \frac{R_B}{\tan(\alpha_{4,B})} \right| \tag{9}$$

Because the limitations of the size of the crease pattern are mostly defined by the largest of these two dimensions, the stride length is normalized as follows:

$$L_s = \frac{\Delta X}{\max(l_{cp}, w_{cp})} \tag{10}$$

3 Results & Discussion

The parametric sweep is iterated two times. The first iteration covers the entire design space in order to map the overall qualitative behavior. A second iteration is performed on an area of interest, in order to more closely examine the local quantitative behavior.

3.1 Overall qualitative behavior

Using parameter step sizes $\Delta R = 0.1$, and $\Delta\alpha = 5°$, a study on the overall behavior of the geometry has been conducted. Out of 3,913,539,300 possible parameter combinations for all four RBMs combined, a total of 26,872 paths have emerged that can be considered feasible for constant-height walking.

Because the intended design case utilized four-leg units, the contact ratio C should be greater than 0.25 in order to maintain constant contact with the object. Since RBM_{FT} allows for nearly 98% of the feasible paths with $C > 0.25$, the main body of this paper focuses mostly on this Rigid Body Mode. A full reference work containing the characteristics of the remaining RBMs can be found in Appendix E.

As shown in Figure 6b, three classes of geometries exist in distinct regions of the design space. These three classes exhibit different qualitative behavior by folding differently, resulting in three distinct types of output paths (Fig. 7). Because these classes are bound by the feasibility requirements, sudden changes in criteria scores can be spotted when studying the design space.

The fold patterns of Class I and II are relatively similar, with the primary difference in their behavior arising from the difference in sector angles around vertex v_3. Geometries in Class I have a narrow sector angle $\alpha_{3,F}$, resulting in a steep downward fold of facets F_E and F_D. In contrast, Class II, this sector angle is more moderate, leading these facets to fold slightly upward. Despite the significant difference in $\alpha_{3,F}$, the sum of the three variable sector angles around v_3 remains roughly equal between the two classes, resulting in a similar $\alpha_{3,D}$, which subsequently leads to a similar $\alpha_{4,D}$. Because the sector angles around v_4 also remain similar between the two groups, the main difference in output motion is caused by the different transmission of the input angles towards the fold angle ρ_{3-4} of the crease line connecting the two internal vertices. This is in turn translated to a difference in fold angle ρ_{4-7}. This difference leads to Class I having a lower, wider output path (Fig. 7, resulting in a higher average step size L_s and a higher contact ratio C.

Moreover, our findings demonstrate that Class II and Class III have similar dimensions and behavior around vertex v_3. However, their behavior around vertex v_4 is significantly different. Where in crease line c_{4-7} folds in mountain fold configuration for Class I and Class II, its orientation is reversed for Class III. As a result, the horizontal section of the output path is located at the top, in contrast to the other two classes.

Because the output velocity during the propulsion phase is significantly higher for Class III, these paths have a relatively low contact ratio, and are therefore not suitable for the four-legged active surface units. Additionally, due to the inward folding of all facets in Class III, mechanisms in this class are susceptible to self-intersection. Therefore its self-intersection is highly dependent on the auxiliary parameters, which often results in unpractical dimensions. This makes this class difficult to implement in a physical application.

3.2 Quantitative behavior in Class I

As can be seen in Figure 6, Class I is the only class that exhibits a sufficient contact ratio to be implemented in the four-legged unit system. Furthermore, it also scores highest on other criteria, as is shown in Appendix E. Therefore, this region is used to define a new design space, specifically for the intended design case. In this region, a higher resolution parametric study can be repeated in order to further analyze the quantitative difference between solutions inside this class, which are used to find an optimized design. In this second study, the step sizes are halved to $\Delta\alpha = 2.5°$, and $\Delta R = 0.05$, which has resulted in 9561 newly generated feasible paths for RBM_{FT}. Figure 8a shows a comparison of the two error criteria ϵ_z and ϵ_s of all feasible outcomes of this second study, together with their stride length L_s.

As shown in Figure 8, there exists an approximately inverse relation between the error in X-velocity ϵ_v and the stride length L_s. A strong correlation exists between L_s and contact ratio C, as a larger stride length is often a result of a longer contact ratio.

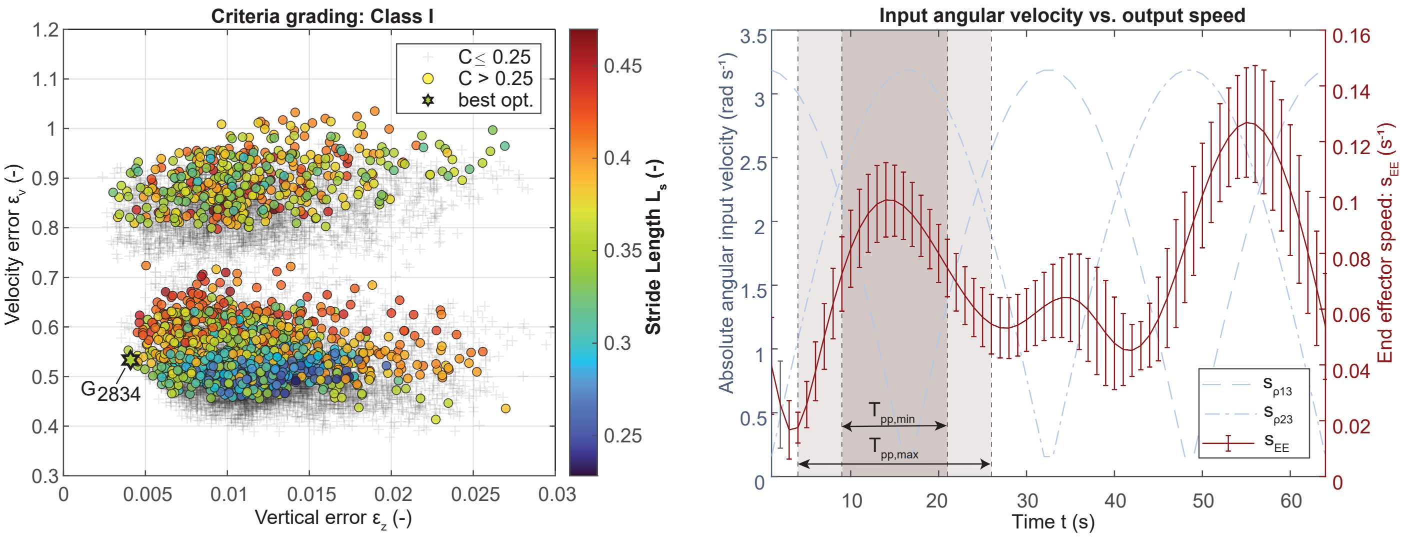


Figure 8: a) Scatter plot showing the criteria grading of all feasible paths in the narrowed down design space of Class I, b) Plot showing the relation between the input angular velocities ($s_{\rho 13}$ & $s_{\rho 23}$) and the output speed s_{EE}, together with an indication of the maximum and minimum time intervals of the propulsion phase.

Furthermore, contact ratio C is inversely related to ϵ_v. This inverse correlation can be explained by the fact that the speed profile of the end effector over time remains roughly constant for the chosen family of geometries. As can be seen from the output path in Figure 7a, a large speed difference is present around the transition between the propulsion phase and return phase. As shown in Figure 8b, when the contact ratio increases, the propulsion phase covers a larger segment of this speed difference. Therefore, a mechanism with a large stride length will generally suffer from a larger ϵ_v.

Therefore, with the current input, a compromise must be made between stride length and constant propulsion velocity. Because the selected geometries with a contact ratio of $C > 0.25$ have a relatively large step size, and eliminating hysteresis is important in the intended application, ϵ_v is prioritized, resulting in the following optimized geometry:

$$R_2 = 2.10$$
$$\alpha_{3,F} = 15^\circ$$
$$\alpha_{3,E} = 97.5^\circ$$
$$\alpha_{3,D} = 132.5^\circ$$

$$\alpha_{4,D} = 90^\circ$$
$$\alpha_{4,B} = 27.5^\circ$$

3.3 Auxiliary parameters

Once the primary parameters have been established, the other parameters of the crease pattern can be chosen. However, in order to prevent the mechanism from colliding with the locomotion plane or intersecting with itself, the values of some of these auxiliary parameters are bound.

Because they fold in the direction of the locomotion plane, the maximum length of crease lines c_{3-5} and c_{4-9} is limited such that they do not collide with the plane at their steepest angle. In addition, the maximum size of the facets adjacent to these creases is derived by analyzing the angle between the crease lines and the intersection of these facets with the locomotion plane, at their most critical positions. This results in the following limitations, as displayed in Figure 9:

$$L_{3-5} < 1.08R_A$$
$$L_{4-9} < 2.10R_A$$
$$\beta_{5,E} < 85.4^\circ$$
$$\beta_{5,D} < 107.9^\circ$$
$$\beta_{9,D} < 104.2^\circ$$
$$\beta_{9,C} < 96.4^\circ$$
$$\beta_{8,C} < 99.4^\circ$$

The dimensions of the remaining crease lines have no risk with colliding with the locomotion plane as they are either fixed horizontally (c_{1-3}, c_{2-3}, and c_{4-7}), or are pointed upwards (c_{3-6}). Therefore, they are free to choose according to other design requirements such as mechanism size, load bearing capacity, or dynamic properties.

3.4 Prototyping and evaluation

In order to physically validate the computational model, a prototype was built following a similar method as Melancon et al. [28]. The crease line pattern is laser cut into two 400 g paper sheets. Subsequently, the created paper-based facets are aligned and placed on both sides of double-sided adhesive tape, resulting in a sandwich structure with relatively stiff facets, and relatively compliant creases. The best performance is measured with a 0.8 mm crease width. More details on the fabrication of the prototype can be found in Appendix F.

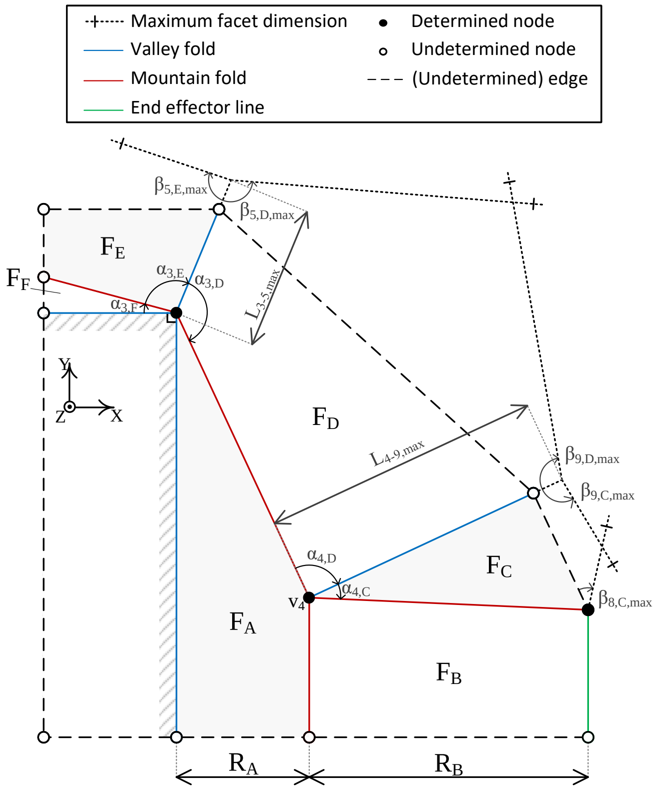


Figure 9: Proposed crease pattern of the optimized geometry, with its primary parameters, maximum dimensions, and crease pattern assignment determined.

After the mechanism is clamped onto an elevated platform, the X- and Z-components of the end-effector position are measured. These measurements are performed relative to the origin, for 16 distinct instances of the motion. Subsequently, their exact position is determined using an image processing script created with MATLAB, which is then compared to the path generated by the computational model. As shown in Figure 10b , the physical model partly confirms our findings. However, a relatively large error is present, which causes the end effector to deviate from the intended straight horizontal section.

The error mainly results from two factors. Firstly, when the dihedral are high (i.e. when the X-coordinate is close to zero), the non-idealized stiffness in the crease lines cause the mechanism to deform. Since the largest fold angles are concentrated around v_4, the creases around this vertex unfold slightly, resulting in an error in the positive X- and Z-directions. Secondly, the weight of the mechanism deforms the finite-stiffness facets, and causes the creases to deform orthogonal to their axes. This causes the position of the end effector to deflect slightly downwards.

Both errors are confirmed by Figure 10c, which shows the correlation between X-coordinate of the end effector and the resulting error. In addition to these phenomena, several other factors contribute to the non-ideal behavior of the prototype. When vertex v_4 folds inwards, large stresses arise when the fold angles of the surrounding creases become too high. This causes facet F_A to deform, which distorts the direction of crease line c_{4-7}. As a result of this phenomenon, the end effector moves out of its intended plane, further distorting its path, and rendering line contact unattainable. An effective solution to this problem is to create a relatively large hole surrounding vertex v_4, which prevents large concentrations of stress.

4 Outlook

While an optimum in the design space has been found, no geometry exists in this design space that has a sufficiently low velocity error ϵ_v in order to create the intended active surface for high precision manipulation. However, the proposed crease pattern, together with the developed methodology and results, can be utilized in the design process of other origami-based mechanisms.

Firstly, the generated data can be used to create other types of walking motion that can be utilized for other applications than an active surface. For instance, an extra criterion can be defined to assess the step height of the legs, which can be used in the development of an origami-based mobile robot that requires the ability to overcome obstacles and elevations.

More generally, the developed crease pattern and methodology can be used to approach other prescribed cyclical output motions, which can be used in other applications than walking. For instance, where current origami-based metamaterials often exist of simple single-vertex unit cells or known tessellated patterns [29], the proposed double-vertex crease pattern can serve as a base for more intricate materials. Furthermore, by extending the crease pattern with more parallel-crease vertices, linkage systems with more than two revolute joints can be created. This can then be combined with conventional theory about planar linkage synthesis to approach more complicated output shapes.

5 Conclusion

This paper has introduced a new crease pattern with parallel crease lines for a 2-DOF origami-based mechanism that can perform a straight-line horizontal motion necessary for constant-height walking locomotion. The presented crease pattern can be used for a locomotion mechanism that requires individual control over the two degrees of freedom, which can be used for robots that can perform relatively complex tasks that require high levels of maneuverability. Furthermore, a coupled input can be used when simpler control is favorable. For this case, the parametric study has found many types of feasible straight-line paths with different characteristics. The results of this parametric study indicate several distinct regions in the design space for which favorable characteristics occur. Subsequently, this has been utilized to optimize the geometry of the crease pattern for a high-precision active surface.

While it has been found that no geometry exists for the presented crease pattern to achieve a locomotion pattern suitable for a high-precision active surface, the results of this study provide valuable insights into the behavior of the presented crease pattern and its potential applications. The created crease pattern can be utilized in the design of various other applications that require different characteristics, while the presented results of the parametric study provides a reference work for future designs.

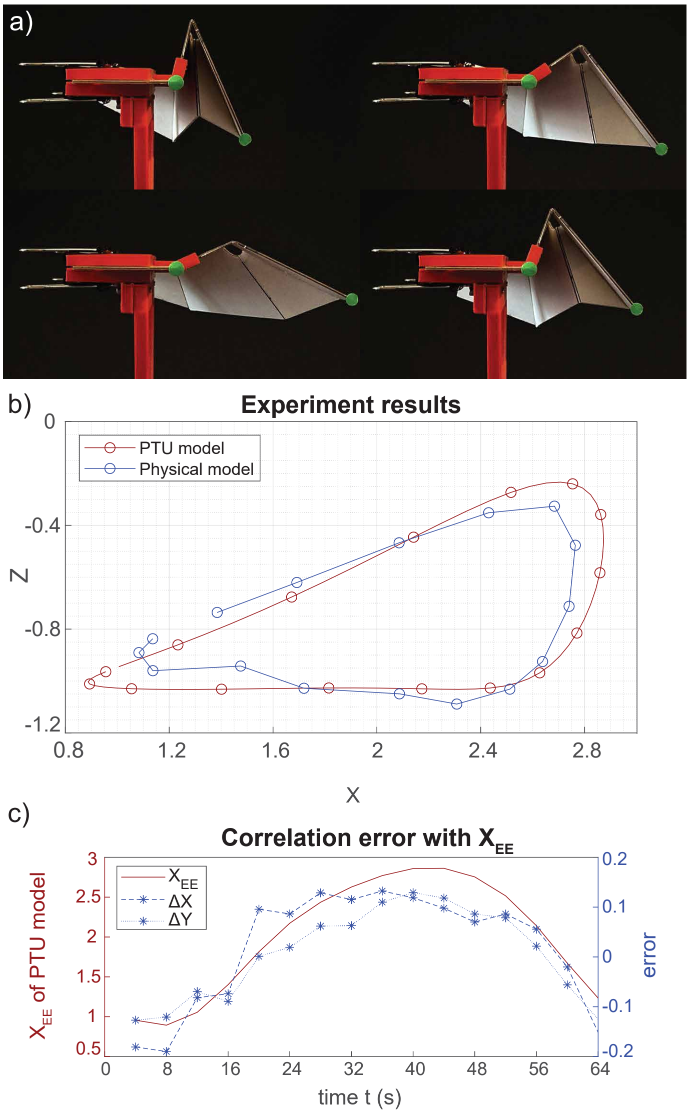


Figure 10: Results of the experimental validation. a) Mechanism position at four of the sixteen time steps, as used by the image processing code. b) Comparison between experimental results and path generated by PTU. c) Correlation between the X-coordinate of the end effector and the error in the experimentation.

References

[1] S. A. Zirbel, B. P. Trease, S. P. Magleby, and L. L. Howell, "Deployment Methods for an Origami-Inspired Rigid-Foldable Array," in *The 42nd Aerospace Mechanism Symposium*, 2014.

[2] "Space Technology Game Changing Development A-PUFFER: Autonomous Pop-Up Flat-Folding Explorer Robots," 1995.

[3] M. Dorigo, G. Theraulaz, and V. Trianni, "Swarm Robotics: Past, Present, and Future [Point of View]," in *Proceedings of the IEEE*, vol. 109, pp. 1152–1165, 2021.

[4] D. Rus and M. T. Tolley, "Design, fabrication and control of origami robots," *Nature Reviews Materials*, vol. 3, pp. 101–112, 2018.

[5] S. F. Miller, C. C. Kao, A. J. Shih, and J. Qu, "Investigation of wire electrical discharge machining of thin cross-sections and compliant mechanisms," in *Proceedings of IMECE04 2004 ASME International Mechanical Engineering Congress and Exposition*, 2004.

[6] J. H. Cho, M. D. Keung, N. Verellen, L. Lagae, V. V. Moshchalkov, P. Van Dorpe, and D. H. Gracias, "Nanoscale origami for 3D optics," *Small*, vol. 7, no. 14, 2011.

[7] K. C. Francis, L. T. Rupert, R. J. Lang, D. C. Morgan, S. P. Magleby, and L. L. Howell, "From crease pattern to product: Considerations to engineering Origami-adapted designs," in *Proceedings of the ASME Design Engineering Technical Conference*, vol. 5B, pp. 1–15, 2014.

[8] A. Vorob'ev, P. Vaccaro, K. Kubota, S. Saravanan, and T. Aida, "Array of micromachined components fabricated using "micro-origami" method," in *Japanese Journal of Applied Physics, Part 1: Regular Papers and Short Notes and Review Papers*, vol. 42, pp. 4024–4026, Japan Society of Applied Physics, 6 2003.

[9] Z. Zhakypov, K. Mori, K. Hosoda, and J. Paik, "Designing minimal and scalable insect-inspired multi-locomotion millirobots," *Nature*, vol. 571, no. 7765, pp. 381–386, 2019.

[10] S. M. Felton, M. T. Tolley, C. D. Onal, D. Rus, and R. J. Wood, "Robot self-assembly by folding: A printed inchworm robot," in *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 277–282, 2013.

[11] J. S. Koh and K. J. Cho, "Omegabot: Biomimetic inchworm robot using SMA coil actuator and smart composite microstructures (SCM)," in *2009 IEEE International Conference on Robotics and Biomimetics, ROBIO 2009*, pp. 1154–1159, 2009.

[12] C. D. Onal, R. J. Wood, and D. Rus, "An Origami-Inspired Approach to Worm Robots," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 2, pp. 430 – 438, 2013.

[13] A. Pagano, T. Yan, B. Chien, A. Wissa, and S. Tawfick, "A crawling robot driven by multi-stable origami," *Smart Materials and Structures*, vol. 26, no. 9, 2017.

[14] E. V. Hoff, D. Jeong, and K. Lee, "OrigamiBot-I: A Thread-Actuated Origami Robot for Manipulation and Locomotion," in *IEEE International Conference on Intelligent Robots and Systems*, (Chicago, IL, USA), pp. 1421–1426, IEEE, 2014.

[15] M. Yu, W. Yang, Y. Yu, X. Cheng, and Z. Jiao, "A crawling soft robot driven by pneumatic foldable actuators based on Miura-ori," *Actuators*, vol. 9, 6 2020.

[16] H. Fang, Y. Zhang, and K. W. Wang, "Origami-based earthworm-like locomotion robots," *Bioinspiration and Biomimetics*, vol. 12, no. 6, p. 065003, 2017.

[17] P. Bhovad, J. Kaufmann, and S. Li, "Peristaltic locomotion without digital controllers: Exploiting multi-stability in origami to coordinate robotic motion," *Extreme Mechanics Letters*, vol. 32, 2019.

[18] W. P. Weston-Dawkes, A. C. Ong, M. R. A. Majit, F. Joseph, and M. T. Tolley, "Towards rapid mechanical customization of cm-scale self-folding agents," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-Septe, pp. 4312–4318, 2017.

[19] J. Li, H. Godaba, Z. Q. Zhang, C. C. Foo, and J. Zhu, "A soft active origami robot," *Extreme Mechanics Letters*, vol. 24, pp. 30–37, 2018.

[20] W. H. Chen, S. Misra, J. D. Caporale, D. E. Koditschek, S. Yang, and C. R. Sung, "A Tendon-Driven Origami Hopper Triggered by Proprioceptive Contact Detection," in *2020 3rd IEEE International Conference on Soft Robotics, RoboSoft 2020*, pp. 373–380, IEEE, 5 2020.

[21] A. T. Baisch, O. Ozcan, B. Goldberg, D. Ithier, and R. J. Wood, "High speed locomotion for a quadrupedal microrobot:," *The International Journal of Robotics Research*, vol. 33, pp. 1063–1082, 5 2014.

[22] A. M. Hoover, E. Steltz, and R. S. Fearing, "RoACH: An autonomous 2.4g crawling hexapod robot," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, (Nice, France), pp. 26–33, 2008.

[23] S. G. Faal, F. Chen, W. Tao, M. Agheli, S. Tasdighikalat, and C. D. Onal, "Hierarchical Kinematic Design of Foldable Hexapedal Locomotion Platforms," *Journal of Mechanisms and Robotics*, vol. 8, no. 1, p. 011005, 2016.

[24] S. Felton, M. Tolley, E. Demaine, D. Rus, and R. Wood, "A method for building self-folding machines," *Science*, vol. 345, no. 6197, pp. 644–646, 2014.

[25] Y. Kim, Y. Lee, and Y. Cha, "Origami Pump Actuator Based Pneumatic Quadruped Robot (OPARO)," *IEEE Access*, vol. 9, pp. 41010–41018, 2021.

[26] L. Zimmermann, T. Stankovic, and K. Shea, "A Computational Design Synthesis Method for the Generation of Rigid Origami Crease Patterns," *Journal of Mechanisms and Robotics*, vol. 14, 6 2022.

[27] L. Zimmermann, K. Shea, and T. Stanković, "Conditions for Rigid and Flat Foldability of Degree-n Vertices in Origami," *Journal of Mechanisms and Robotics*, vol. 12, 2 2020.

[28] D. Melancon, B. Gorissen, C. J. García-Mora, C. Hoberman, and K. Bertoldi, "Multistable inflatable origami structures at the metre scale," *Nature*, vol. 592, no. 7855, pp. 545–550, 2021.

[29] Z. Zhai, L. Wu, and H. Jiang, "Mechanical metamaterials based on origami and kirigami," 2021.

Chapter 2

Further Recommendations

While an optimum in the design space has been found, no geometry exists in this design space that has a sufficiently low velocity error ϵ_v in order to create the intended active surface for high precision manipulation. Further studies could try to improve this through several ways.

Firstly, the design space of the parameters can be expanded. Decreasing the step sizes $\Delta\alpha$ and ΔR will probably not result in any lower speed error, since the behaviour is quite constant. However, while the sector angles already cover the entire possible design space, dimensional parameter R_2 can be expanded. Figure **??** suggests that the optimum of velocity error ϵ_v lies at $R_2 > 3.0$. However, such a geometry will likely have a lower contact ratio, and therefore, more input signals will be needed to operate the active surface.

Secondly, different types of input profiles can be explored. One could use another sinusoidal input with a different range, a different phase shift, or opposite fold directions of the input creases. However, as Figure **??**b suggests, part of the speed difference is a result of a non-constant angular velocity at the inputs. Therefore, a more promising solution would be to use a different input profile, such as a triangle wave, which has a constant angular velocity.

Thirdly, a similar parallel-crease crease pattern with three or more internal vertices can be used in the same fashion. If one would extend vertex v_8 according to the rules stated in the topology synthesis method, one could simulate a RRR planar linkage with a crease pattern that still has two degrees of freedom. This would allow for much more complex motion patterns. However, the computational costs would be a lot higher.

Lastly, one could research the workings of one of the other concepts as presented in Appendix A. For instance, the same concept that prevents the mechanism from entering a singularity, can be utilized to detect singularities between two Rigid Body Modes, such that a circular path can be created with a topology that provides only one degree of freedom.

While the presented geometry will not work for high-precision applications, it can still be used in more low-tech applications, where hysteresis is less important. Further research is needed to develop the chosen geometry, such as studying the possible actuation methods like magnetic actuation of pneumatic pouches [1]. Furthermore, a further studies must be done on facet materials, and methods to create crease lines with low bending stiffness while maintaining an as large as possible stiffness in the other directions in order to decrease the complications that have arisen in the physical evaluation.

Moreover, especially in active surfaces with large numbers of legs, or robots that fold remotely, distributed actuation methods need to be integrated in order to automatically unfold the robot out of its flat-folded singularity state. Lastly, for integration into an active surface, further studies can be performed on creating rotational motion in order to manipulate the object in all three planar degrees of freedom, X, Y, and R_Z. This could be performed by using multiple high-radius curved horizontal paths that have a remote center of rotation. Another method could be to rotate around a single separate unit that performs a circular arch around one of its vertices. Such a unit could be created from a $N_{int} = 1$ topology, as described in Appendix B.

Appendix A

Functional Analysis

This appendix describes the process of the generation, classification, and selection of concepts.

The main purpose - to develop an origami-based mechanism that generates closed-loop output curve with a horizontal line section - can be split into two main functions, which both can be solved by their own distinct strategies. Furthermore, several options exist to use as an input method

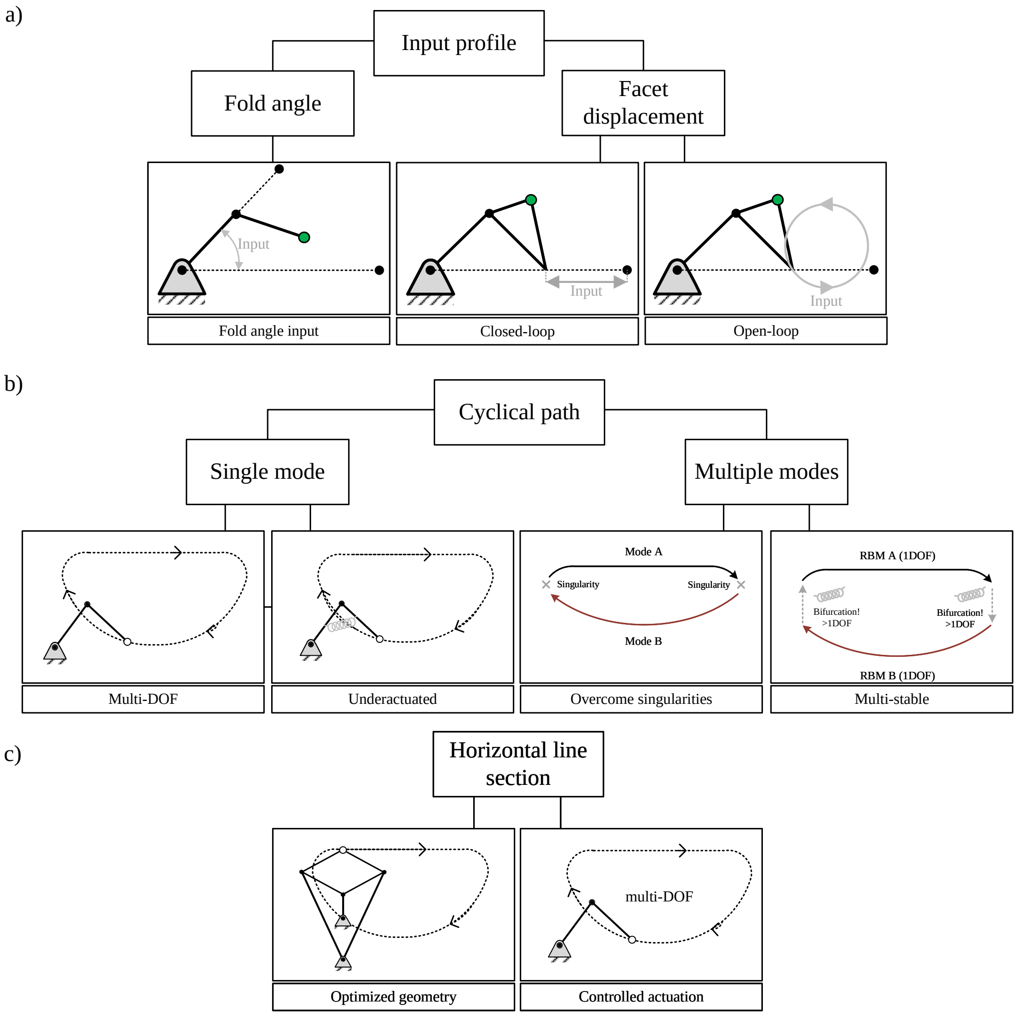


Figure A.1: Overview of different design options used to create the leg mechanism. a) Different input methods. b) Different strategies to solve the function of creating a cyclical path. Categorized by their working principles. c) The two available strategies to create a horizontal line section.

Input

The input signal of an origami-based mechanism can be applied either to the crease line, or to a point on a facet.

When applied to the crease line, the input signal directly controls its fold angle. This type of input can only be in the form of a reciprocal motion. There are many ways to controlling a mechanism by moving a point on its facet. This can be done with either closed-loop or open-loop motion. Figure A.1a shows two examples: a closed-loop linear input in-plane with the origami surface, and an open-loop circular input orthogonal to the surface. However, these inputs can differ vastly in shape and orientation. For instance in-plane closed-loop inputs, and in-plane open-loop inputs could be useful for implementation in an active surface based on lamina emergent mechanisms that have respectively 1-DOF and 2-DOF controlled similar to concepts proposed by Gollnick et al. [2]. While this could be a feasible solution for an active surface, it is out of the scope of this study.

Creating a cyclical path

Firstly, in order for the end-effector to be able to return to its starting point without causing friction or colliding with the propulsion plane, the output path should be cyclical. Crease lines in origami-based mechanisms cannot perform continuous rotational motion, as the fold angle in the creases is restricted to a range of motion of a maximum of $[-180°,+180°]$. This range of motion will be decreased further when facet thickness is taken into account. For origami mechanisms with angular actuation in a crease line, that have a single degree of freedom and operate in only one mode, the relation between the input angle and the output is deterministic (i.e., a certain input angle always results in the same output). Therefore, it is impossible to create a cyclical output for such a system. Consequently, in order to create a cyclical output from a non-cyclical input, the mechanisms should have at least two degrees of freedom, or multiple RBMs should be accessed. Figure A.1b shows an overview of the available strategies.

Multiple degrees of freedom

If the facets are to remain rigid when using multiple degrees of freedom, both DOFs have to be actuated. On the contrary, when compliance in the facets can be exploited, one could design an under-actuated system, such that only a single DOF has to be controlled. To achieve this, elastic potential energy can be stored in the deforming facets during the propulsion phase, which can be released during the return phase, such that the mechanism returns in a path with the most favorable energy profile.

Multiple modes

In order to avoid multiple degrees of freedom, one can access more than one mode of the folding pattern during one cycle.

If multiple Rigid Body Modes (RBMs) are accessed, the origami travels through a state where all crease lines around a vertex are folded flat. In this state, the mechanism is in singularity, thus it can not be actuated properly. If rigid facets are used, momentum can be used to overcome this singularity [3]. Another method to avoid reaching this singularity point is by creating a mechanism with two stable non-rigid modes. This method can be seen as analogous to using multiple Rigid Body Modes, where instead of changing fold directions, the mode is determined by the buckling direction of one or more compliant facets. Therefore, the singularity point is in the unstable region, and can therefore be resolved.

A second approach that uses multiple modes to create a cyclical path is to utilize hidden degrees in compliant facets, in order to switch between two stable Rigid Body Modes (i.e., in the stable states there the facets remain undeformed). Using a technique proposed by [4], hidden degrees of freedom arising from the modes in the compliant facets can arise once enough energy is supplied to the mechanism, allowing the mechanism to switch between these states.

Appendix B

Topology Synthesis

To facilitate the first requirement, to create a cyclical path, a 2-DOF origami mechanisms must be created. Therefore, a topology with two rigid internal degrees of freedom should be found.

%sectionReview on available methods

B.1 Computational Design Synthesis

To create a suitable crease pattern topology for the intended design, we have used the Computational Design Synthesis by Zimmermann et al. [5]. This method utilizes graph theory by representing the internal creases in the crease pattern as graphs. This offers a method to systematically expand an initial crease pattern G_0 with extra crease lines, by using a graph grammar with two possible rules, defined according to rules that follow from the PTU method. Firstly, a vertex in the crease pattern can be expanded with three new outgoing creases, forming a new vertex triangle (r_1). As a complementary rule, two existing neighboring vertices can merge to form a new vertex, still satisfying the Principle of Three Units (r_2). By systematically applying these rules, the entire available design space can be searched in a systematic way. The topology generation is bound by a maximum number of internal vertices N_0, which can be defined by the user.

B.1.1 Utilization of the matlabPTU framework

This method has been reproduced and adjusted in several ways in order to better meet our design requirements. Firstly, the method is adjusted to support multi-DOF crease patterns. Moreover, the filters that are originally applied to the generated graphs are appended, in order to mark graphs that can be used to create a straight-line section.

In order to facilitate these adjustments, the matlabPTU framework created by Andreas Walker [] is utilized. In this framework, an `origami.m` class in MATLAB is created, which covers the graph representation of the crease pattern, as well as the transformation from graph to an origami geometry. Furthermore, the PTU method is utilized in order to facilitate the calculation of the folded state of the generated origami crease pattern. The framework of matlabPTU also offers an optimization method that optimizes the topology together with the geometry of the crease pattern in order to approach a prescribed motion or shape. However, this optimization method converges to the nearest local optimum. In order to be able to study all possible outcomes, and in turn find the best outcome for our design objective, we therefore implement the topology generation algorithm as designed by Zimmerman.

Adjustments

The method is reproduced and has been adjusted in several ways, in order to better meet our design requirements. The filters that are originally applied to the resulting graphs are appended, in order to only select graphs that are feasible for the intended application.

Facilitating multiple DOFs

While the original algorithm is utilized for 1-DOF mechanisms, it can simply be adjusted to create crease patterns with multiple degrees of freedom.

a) 2-DOF Initial Graph – The most simple way to generate a 2-DOF topology is to change the initial graph to a 2-DOF input (Fig. B.1a) This will yield the same exact same results as the original algorithm, as this has no effect on the graph grammar rules.

b) Adding an extra crease at the end – Furthermore, one can run a 1-DOF graph grammar, and simply add an extra crease at the end to increase the system with an extra degree of freedom. This crease can be added between two boundary vertices, essentially creating an in-series actuated mechanism (Fig. B.1b), where the extra input does not affect the rest of the mechanism (as its sector angle is no input to any internal vertex. Moreover, one can add an extra crease to a boundary facet that connects to an internal vertex, adding a single input into the graph network (Fig. B.1c). A third option is to add a crease to an internal facet, which adds two equal inputs to two sections of the graph (Fig. B.1d).

c) Counting DOFs – Lastly, a one could adjust the algorithm such that, during the generation, one instance is allowed where a vertex is expanded with a third rule r_3, which expands the vertex with four crease lines, and thus allowing an extra degree of freedom to occur. One of these four crease lines should be actuated such that the system does not undetermined. If the crease is actuated, it functions as extra input to the PTU calculations, and thus guarantees that the mechanism stays determined. This method can yield extra results compared to adding an extra create after the generation, because extra branches can be created (Fig. B.1e).

Note that the results created with method (b) are a subset of the results created with method c), when rule r_2 is used. In turn, the result of method a) is a subset of the results of the other methods (Figure B.1f)

For the particular application of designing a walking locomotion system, a 2-DOF initial graph is used. This guarantees that the actuated crease lines remain fixed to the base enabling purely parallel actuation, which prevents vulnerable connections between actuators, and prevents potentially heavy actuators to move around on the mechanism and impact its dynamics. Note that the topology created with this method can also be achieved with both other methods.

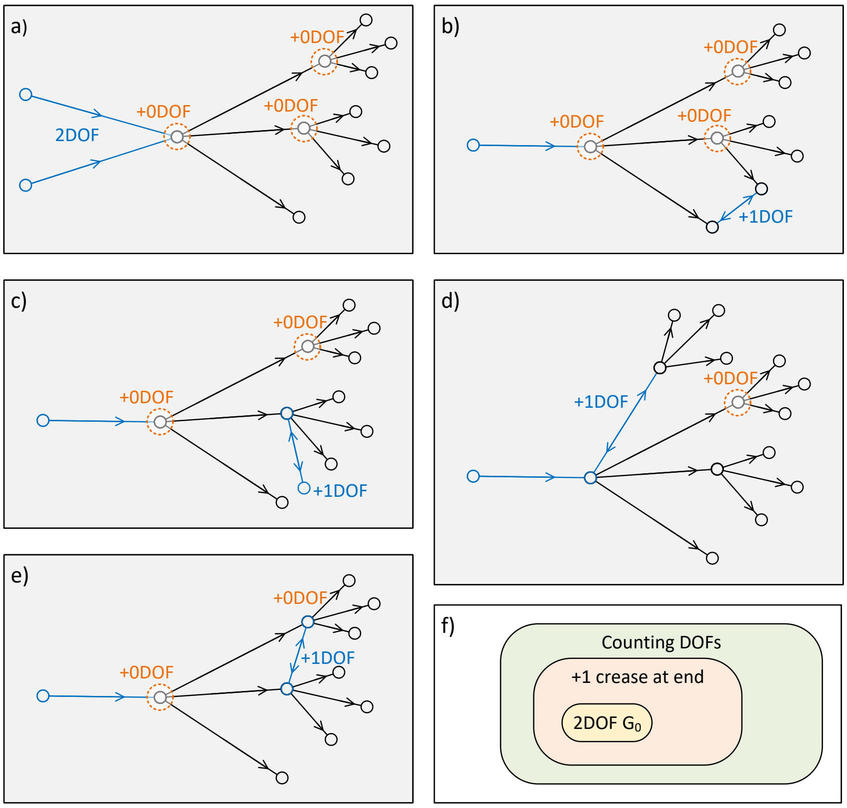


Figure B.1: Descriptions of strategies of altering the graph grammar to generate a 2-DOF crease pattern. a) Using a 2-DOF initial graph G_0. b) Adding an extra crease between two boundary vertices. c) Adding an extra crease to an internal vertex on a boundary facet. d) Adding an extra crease between two internal vertices. e) Allow one vertex extension with four creases. f) An overview of the solution sets of each method.

Selection of end effectors

To select possible end effectors, the crease pattern model created by Walker et al. [6] is used, in contrast to the purely graph-based model by Zimmerman. This allows to select more end effectors, as the creases are complemented with boundary creases on which auxiliary nodes are positioned. Each of these auxiliary nodes can be interpreted as representing an arbitrary position of the facet it is positioned on. Therefore, now any point in the mechanism can be considered as an end-effector. In fact, each of the auxiliary nodes can approach any of the end effectors placed on the boundary nodes, as previously selected by Zimmerman. Therefore, in order to avoid redundant calculations, only the auxiliary nodes are considered as candidates.

Each of the auxiliary nodes is evaluated on their eligibility by subjecting them to a set of requirements. If a node fails to satisfy one of these requirements, it is no longer considered a candidate to be an end effector. If no end-effector candidates remain for a certain topology, it is deemed infeasible. However, it still stored in the generation, as their successors can in fact be considered feasible.

Semantic validity – While the original paper examines the semantic validity of an entire topology, this process is extended such that the semantic validity of every auxiliary node is determined.

Since simplification of the crease pattern is important, an end effector candidate is only considered semantically valid if all nodes in the system affect its behaviour, otherwise, another, more simple crease pattern topology can perform exactly the same, but with fewer creases. As Walker states [6], the behaviour of each node can only be affected by the position and sector angles of their predecessors (and therefore by the position of the direct successors of these predecessors. Therefore, an auxiliary vertex is only considered an end effector candidate if all internal nodes are present in its list of predecessors.

Straight-line capability – Since the objective of the mechanism is to generate a straight horizontal line section, a second requirement is set to the end-effector candidates to rule out nodes that cannot reach this objective regardless of geometry. A vertex in an origami crease pattern can be considered analogous to a spherical joint. When a node has a rigid connection with a vertex, the distance between this node and the vertex is fixed. Therefore, the workspace of the nodes neighbouring a vertex is fixed to a spherical surface around this vertex, where the radius of the sphere is equivalent to the distance between the node and the vertex. Consequently, any node that directly neighbours fixed vertex v_3 is unable to perform a straight-line motion and will be discarded to serve as end effector. However, in future studies, this property could be utilized in the design of a leg that performs a turning motion around the Z-axis emerging from vertex v_3.

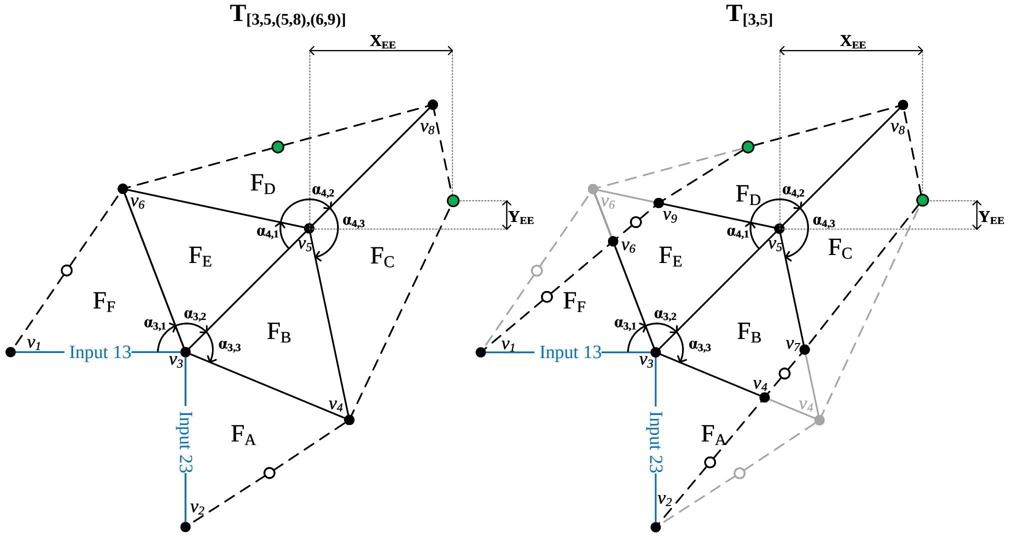


Figure B.2: Two parametrically equivalent topologies. $T_{[3,5,(5,8),(6,9)]}$ is created by merging node pairs v_5-v_8 and v_6-v_9 of topology $T_{[3,5]}$. As shown on the right, this merged topology can be represented as a geometrically limited version of its predecessor.

Parametric equivalence

Besides the filters that the original authors apply to eliminate mirrored and isomorphic graphs, we apply an extra filter that discards topologies with parametric equivalence as well. When topologies have one or more boundary nodes merged according to the graph grammar, and these boundary nodes are not expanded, the resulting graphs are parametrically equivalent to the graph they originate from. They can still be described by the exact same parameters, and therefore, the behaviour of the end-effector is not affected, as shown in Figure B.2. The resulting merged topology does however have extra limitations on its geometry, as the crease lines pointing towards the merged node cannot diverge. Therefore, this topology will be discarded, while the original, unconstrained topology is maintained. However, if the merged node is expanded, the successors of the parametric equivalent topology can be unique.

B.1.2 Reproduction of the graph grammar

The flowchart as shown in Figure B.4 describes the algorithm used to mimic the graph grammar as made by Zimmerman. The full MATLAB script describing this graph grammar can be found in `graphGrammar.m`. The graphs are generated and stored in sequences of generations. To create a generation, all graphs of the previous generation are analyzed and expanded where possible. This is done by analyzing each boundary node of the graph if rule $r1$ or $r2$ can be applied. Only a single rule is applied at each time, resulting in a separate new graph stored in the new generation. The algorithm stops when the used-applied stop-conditions are reached after the last generation is analyzed, i.e., there exists no graph in the last generation on which either rule $r1$ or $r2$ can be applied. The graphs the graph grammar generates when subjected to an initial graph G_0 consisting of a single crease have been compared according to the results as presented by Zimmerman [5] in order to confirm the graph grammar.

B.2 Results

Initially, a more simple crease pattern is desirable. Firstly, a crease pattern with fewer internal nodes requires fewer parameters in the parametric sweep. Where the end effector conditions in a 2-DOF crease pattern with one internal vertex can be described with five independent parameters provided the angle in G_0 is fixed), each additional vertex expansion adds four parameters. Since, if the resolution is fixed, the algorithm scales with $\mathcal{O}(c_p^n)$, minimizing the number of parameters provides computational capacity for a more thorough study with higher resolution. Furthermore, a crease pattern with fewer folds requires fewer calculations to evaluate, and is easier to analyze. Moreover, mechanisms consisting of fewer creases can be considered more robust, and have fewer distortions, as deformation in the imperfect creases is a large contribution to inaccuracy.

Since a more simple crease pattern is desirable, we limit the amount of internal nodes to $N_{int} < 3$. Furthermore, as all nodes that directly neighbour fixed vertex v_3 cannot be considered end-effector candidates, the topology with $N_{int} = 1$ is deemed infeasible, leaving us with two unique solutions: $T_{3,4}$, where a vertex on the side is expanded, and $T_[3, 5]$, where the middle vertex is expanded. In the unlikely case that no solutions arise from these two topologies, a further study can be done on the more complicated crease patterns with $N_{int} \geq 3$.

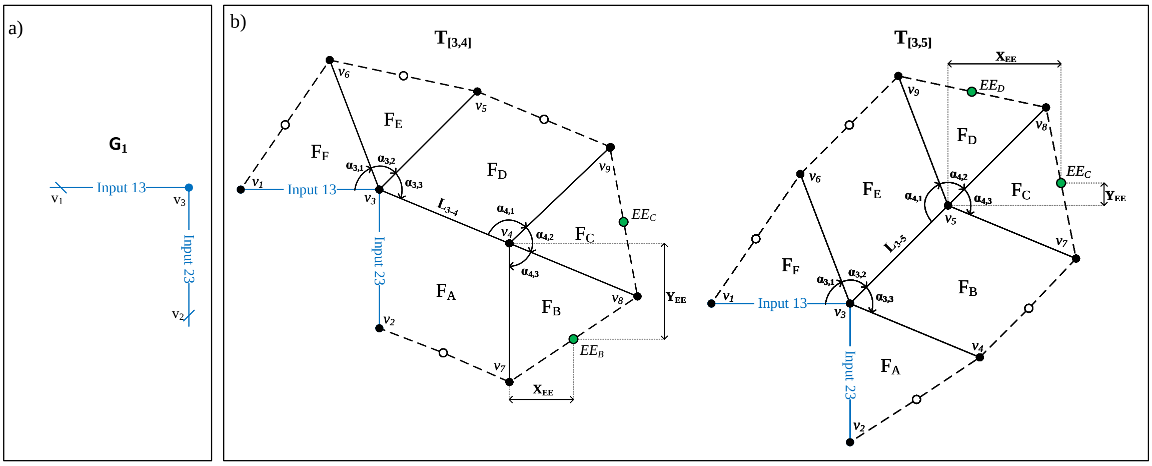
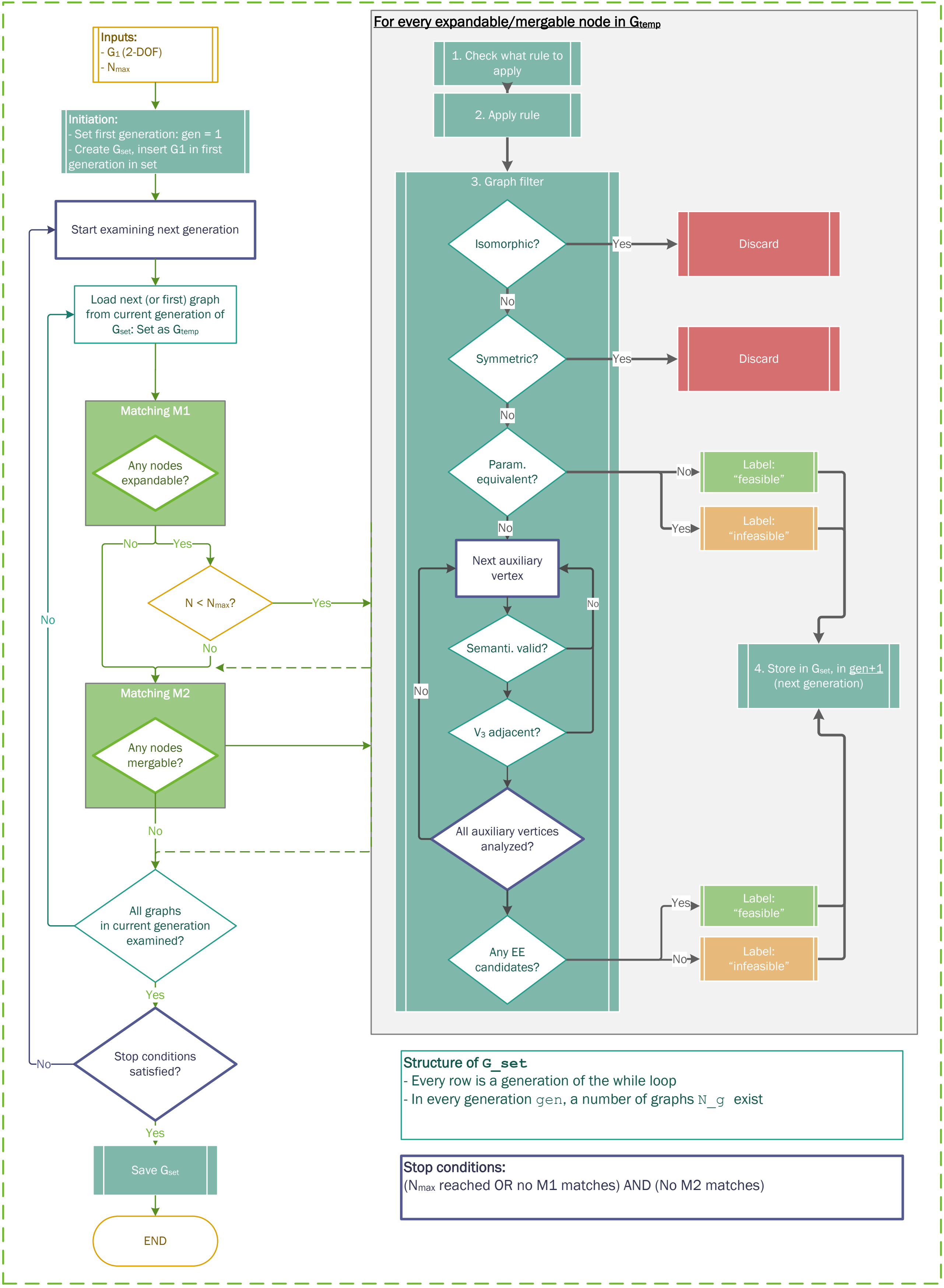


Figure B.3: a) Initial graph with two actuated degrees of freedom G_0. b)Results of the topology synthesis method

Graph Grammar


Inputs:
- G_1 (2-DOF)
- N_{max}
Initiation:
- Set first generation: gen = 1
- Create G_{set}, insert G1 in first generation in set
Start examining next generation
Load next (or first) graph from current generation of G_{set}: Set as G_{temp}
Matching M1
Any nodes expandable?
No
Yes
$N < N_{max}$?
Yes
No
Matching M2
Any nodes mergable?
No
No
All graphs in current generation examined?
Yes
Stop conditions satisfied?
No
Yes
Save G_{set}
END
For every expandable/mergable node in G_{temp}
1. Check what rule to apply
2. Apply rule
3. Graph filter
Isomorphic?
Yes
Discard
No
Symmetric?
Yes
Discard
No
Param. equivalent?
No
Label: "feasible"
Yes
Label: "infeasible"
No
Next auxiliary vertex
No
Semanti. valid?
V_3 adjacent?
No
All auxiliary vertices analyzed?
Any EE candidates?
Yes
Label: "feasible"
No
Label: "infeasible"
4. Store in G_{set}, in gen+1 (next generation)

Structure of G_set
- Every row is a generation of the while loop
- In every generation gen, a number of graphs N_g exist

Stop conditions:
(N_{max} reached OR no M1 matches) AND (No M2 matches)

Figure B.4: A full overview of the graph grammar used to reproduce the topology synthesis.

Appendix C

Adjustments to matlabPTU

This appendix gives a brief overview of the adjustments made to the matlabPTU source code in order to enable its use in the topology generation, facilitate 2-DOF inputs, and to generate the workspace plots required to perform the workspace analysis. The resulting adjusted origami class can be found under `origami.m`.

C.1 Added class properties

Several additional properties have been added to the origami class. This section gives an overview of the added classes, together with their objective.

- `label`: a string that is used to track the labels of the generated topologies
- `M1`: a vector indicating which vertices qualify for rule $r1$
- `M2`: a $n \times 2$ matrix indicating which vertices qualify for rule $r2$
- `gen`: the generation in which the current crease pattern is generated
- `g`: the index of the current crease pattern in the generation
- `grippingCandidates`: vector indicating the vertices that qualify for being an end effector
- `vertexToGripWith`: a scalar indicating the current end effector; used for plotting

C.2 Added methods

- `matching1`
- `rule1`
- `r11`
- `matching2`
- `rule2`
- `updateGrippingCandidates`
- `findGrippingCandidates`
- `allPredecessors`
- `drawRangeOfMotionGIF`
- `drawRangeOfMotion`
- `animateEEpath`

Appendix D

Parametric Study

This appendix describes further information on the working of the algorithm used to perform the parametric study. Furthermore, it highlights some of the measures taken in the algorithm in order to improve its computational performance. The full set of MATLAB codes is provided in the supplementary materials.

D.1 Physical validity

In this stage of the design phase the generated geometries are assessed based on requirements, to confirm a path can be created with a rigid facet origami facet in the given configuration.

Input signal

As described, the input signal consists of a multi-phase sinusoidal signal, where the phase shift of $\Phi = +\frac{pi}{2}$ is applied to input ρ_{2-3}:

$$\rho = \frac{80^\circ + 15^\circ}{2} + \frac{80^\circ - 15^\circ}{2}\cos\left((t-1)\frac{2\pi}{n_t}\right) \tag{D.1}$$

For the first steps of the parametric sweep, where the physically valid and feasible paths are assessed, the input signal is of relatively low resolution, with 16 input steps per cycle $n_t = 16$. In this stage of the algorithm, it is mostly important to minimize the number of calculations to increase the algorithms efficiency. A resolution of $n = 16$ is sufficient to accurately filter out most non-rigidly foldable or intersecting paths, while it also suffices for the assessment of minimal straight-section length, as short straight sections not picked up by the algorithm are too short to be considered feasible altogether. In the evaluation stage however, a higher resolution is important, while the set to be analyzed is significantly smaller. Therefore all remaining paths are recalculated with a resolution of $n = 64$.

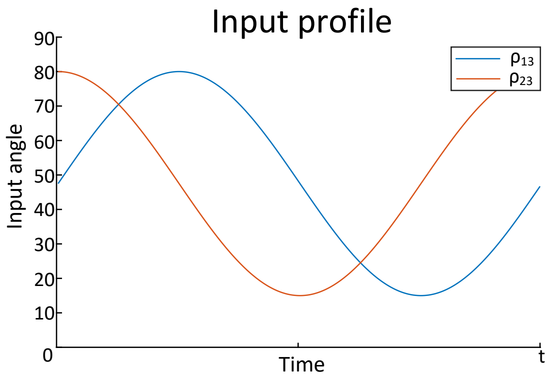


Figure D.1: The input profile of a single unit mechanism

Calculating end effector coordinates

While the PTU method employs 3D vector calculations, and calculates the position of all vertices in the crease pattern, the parametric study can be performed by only calculating the necessary vertex v_8, which represents the location of the end-effector. This greatly reduces the amount of calculations needed, and therefore improves the performance of the algorithm, allowing a study with higher resolution.

As mentioned, since the movement of the end-effector is limited to the XZ-plane, a 2D trigonometric calculation suffices.

This 2D computation reduces the calculation of v_8 from the following system of equations:

$$\begin{aligned}
\mathbf{N}_A &= \mathbf{R}(-\mathbf{v}_{2-3}, \rho_{2-3})\mathbf{N}_Z && \text{(D.2)}\\
\mathbf{v}_{3-4} &= -\mathbf{R}(\mathbf{N}_A, \alpha_{3,A})\mathbf{v}_{2-3} && \text{(D.3)}\\
\mathbf{v}_{4-7} &= -\mathbf{R}(\mathbf{N}_A, \alpha_{4,A})\mathbf{v}_{3-4} && \text{(D.4)}\\
\mathbf{N}_B &= \mathbf{R}(\mathbf{v}_{4-7}, \rho_{4-7})\mathbf{N}_A && \text{(D.5)}\\
\mathbf{v}_{4-8} &= \mathbf{R}(\mathbf{N}_B, \alpha_{4,B})\mathbf{v}_{4-7} && \text{(D.6)}\\
\begin{bmatrix} X_{EE} \\ Y_{EE} \\ Z_{EE} \end{bmatrix} &= \mathbf{v}_{3-4} + \mathbf{v}_{4-8} && \text{(D.7)}
\end{aligned}$$

where rotation matrix $\mathbf{R}(\mathbf{v}, \theta)$ represents a rotation of θ degrees around vector $\mathbf{v}$, to:

$$\begin{bmatrix} X_{EE} \\ Z_{EE} \end{bmatrix} = R_1 \begin{bmatrix} \cos(\rho_{23}) \\ \sin(\rho_{23}) \end{bmatrix} + R_2 \begin{bmatrix} \cos(\rho_{23} + \rho_{47}) \\ \sin(\rho_{23} + \rho_{47}) \end{bmatrix} \tag{D.8}$$

Intersection between non-adjacent facets

Many existing algorithms exist that can be used to detect intersection of planes in 3D space. The Triangle-Triangle Intersection method by Moeller [7] is a commonly used algorithm used in origami-based calculations. However, since only the primary parameters are determined in this stage of the design phase, and consequently the facet dimensions are not yet established, the risk exists that such an algorithm discards geometries that would otherwise have been considered valid. Furthermore, while the Triangle-Triangle method is efficient when compared to other plane intersection methods, the algorithm has to be applied to every non-neighbouring facet combination, which makes applying the algorithm computationally expensive. By examining the behaviour of the the mechanisms, it is found that the vast majority of non-neighbouring facet intersections is caused by the collision of facets F_D, and F_E with the base facet. Therefore, a highly efficient filter is created that implements a simple condition on crease c_{3-5}, where it cannot traverse trough the fixed facet. This discards the majority of the infeasible geometries. The remaining intersecting geometries can often be solved by adjustment of the auxiliary parameters, or can be manually discarded later in the process.

Optimized order of operations

While the parametric sweep varies over all parameters, not every calculation has to be performed for every parameter combination possible. Firstly, many calculations only consider the sector angles around vertex v_3, and therefore, these calculations can be taken out of the for-loop, to avoid repetitions. This regards the calculation of parameters such as remaining sector angle $\alpha_{3,A}$ and the length of crease line c_{3-4}, as R_A is considered constant. Therefore, the geometric validity of geometries with identical values for the sector angles around vertex v_3 only has to be calculated once. More importantly, the same principle can be applied to the PTU calculations around this vertex, which are significantly more computationally extensive. Furthermore, this change of order can be exploited to detect intersections between the facets in an early stage, discarding many geometries even before their end-effector position is calculated. Further adjustments in the order of operations are made such that parallel computing can be used, as facilitated by the MATLAB Parallel Computing Toolbox, and in order to facilitate better memory management of the generated data. Because for higher resolutions the algorithm generates too much data to be stored in RAM memory, the order of operations is changed such that subsets of the data can be intermediately stored on the hard disk, to be used further on in the calculations.

D.2 Feasibility

Straight-line section

As the input signal in the algorithm is discrete, the straight-line sections are detected by studying the existence of horizontal segments between the time steps. Since Z-coordinate of these sections does not matter, they are detected by calculating the finite difference between the time steps. Because in this stage of the design phase, the mechanism size is not considered, the length of a section is determined by the number of horizontal segments in a row. In case the output path contains two separate straight-line sections (e.g., one on top of the curve, and one on the bottom), these paths are split, and both considered as a separate geometry in the rest of the algorithm.

Collision with vertex v_4

In this stage of the design phase, the only part of the mechanism that has a determined location, apart from the end-effector and the fixed base, is vertex v_4, and therefore vertex v_4 is the only vertex that qualifies for a collision check with the locomotion plane. Because of the fact that the input is fixed, and the parameters of the crease pattern are defined such that R_A is constant, the maximum location of vertex v_4 is the same for every examined geometry:

$$Z_{4,max} = R_A \sin(\rho_{2-3,max}) = R_A \sin(80^\circ) \tag{D.9}$$

Therefore, a collision only occurs when the Z-coordinate of the locomotion plane is both positive, and its absolute value is lower than $Z_{4,max}$. Collisions with other, not yet determined parts of the mechanisms can often be avoided by proper adjustment of the auxiliary parameters. However, while this strategy discards the majority of colliding mechanisms, there is a possibility that a geometry folds such that a collision is not to be avoided unless unrealistic auxiliary geometries are applied. In these cases, a solution must be discarded after manual inspection in a later stage of the design process, and the next best solution must be selected to continue the design.

In order to greatly improve the efficiency of the algorithm, the calculation of $Z_{4,max}$ is performed in advance, and it is prioritized in the order of operations, such that it filters out the infeasible paths before other, more extensive, computations are performed.

Appendix E

Additional Results

This appendix shows the full results for all Rigid Body Modes of the comprehensive parametric sweep, which can be used as a reference work for designing a locomotion system for a different type of application.

E.1 Full overview of heatmaps

Besides the number of feasible generated paths, the classes in which these can be categorized, and the contact ratio displayed in the main paper, this appendix provides an overview of heatmaps for all other criteria proposed in the paper, as well as an extra criterion for step height, which is used to complete a second design case.

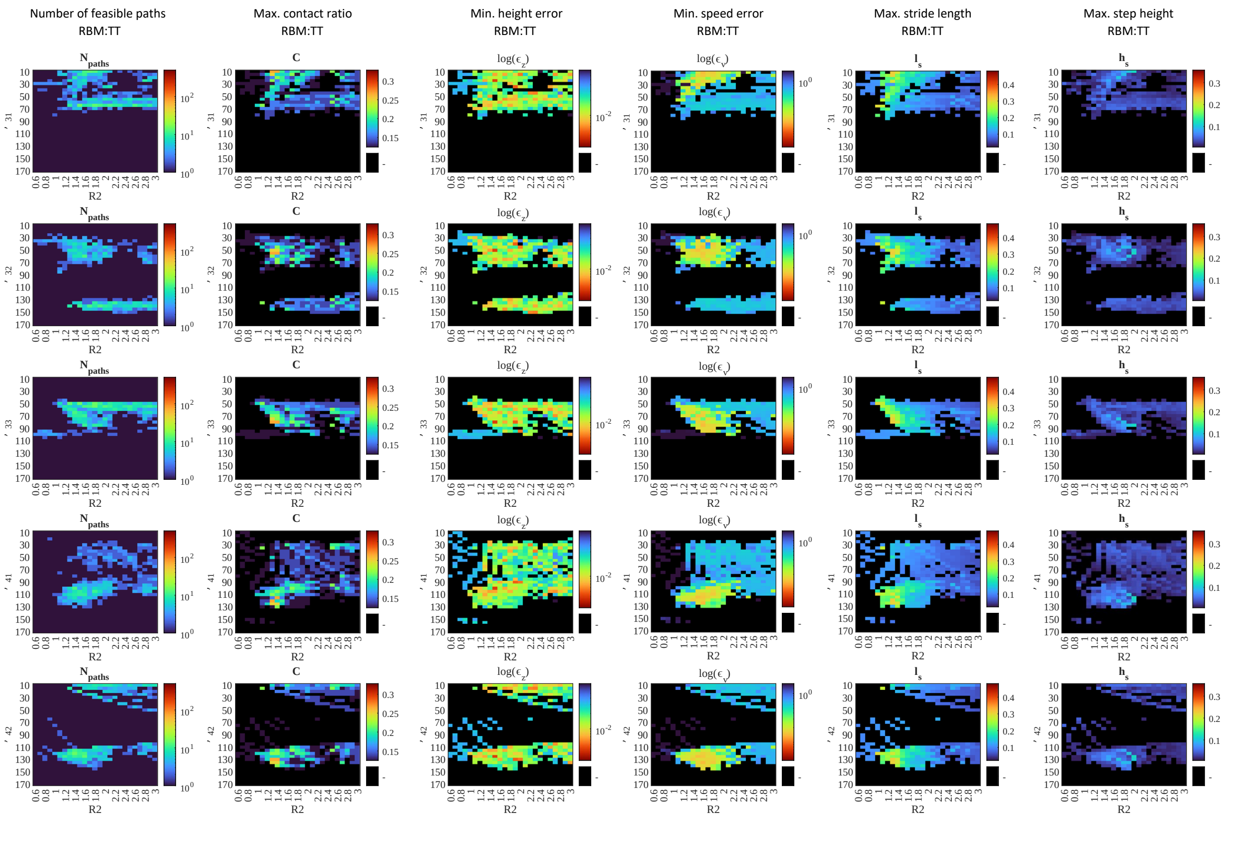


Figure E.1: Overview of all heat maps describing the most favorable criteria ratings for each geometry.

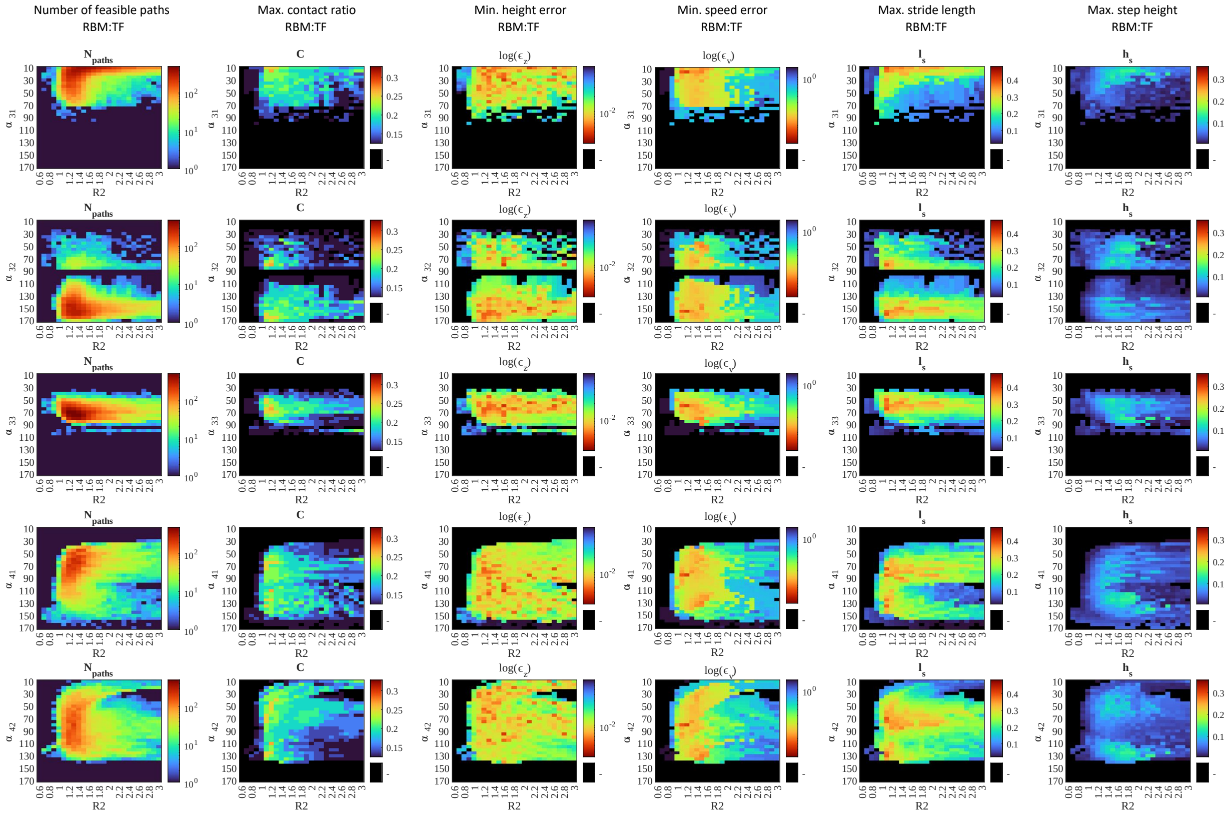

Number of feasible paths
RBM:TF
Max. contact ratio
RBM:TF
Min. height error
RBM:TF
Min. speed error
RBM:TF
Max. stride length
RBM:TF
Max. step height
RBM:TF
N_{paths}
C
$\log(\epsilon_z)$
$\log(\epsilon_v)$
l_s
h_s
α_{31}
α_{32}
α_{33}
α_{41}
α_{42}
10 30 50 70 90 110 130 150 170
R2
0.6 0.8 1 1.2 1.4 1.6 1.8 2 2.2 2.4 2.6 2.8 3
10^0 10^1 10^2
0.15 0.2 0.25 0.3
10^{-2}
10^0
0.1 0.2 0.3 0.4
0.1 0.2 0.3
-

Figure E.2: TF

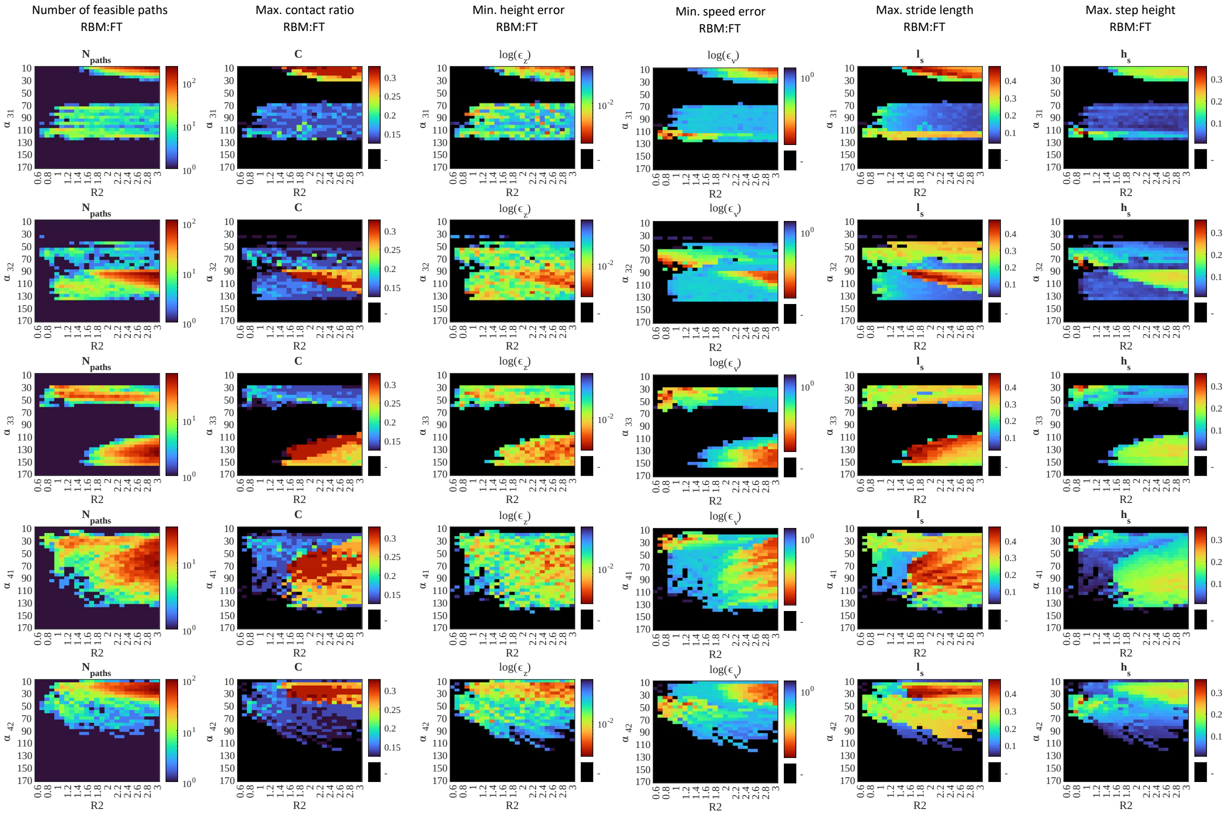

Number of feasible paths
RBM:FT
Max. contact ratio
RBM:FT
Min. height error
RBM:FT
Min. speed error
RBM:FT
Max. stride length
RBM:FT
Max. step height
RBM:FT
N_{paths}
C
$\log(\epsilon_z)$
$\log(\epsilon_v)$
l_s
h_s
α_{31}
α_{32}
α_{33}
α_{41}
α_{42}
R2
10 30 50 70 90 110 130 150 170
0.6 0.8 1 1.2 1.4 1.6 1.8 2 2.2 2.4 2.6 2.8 3
10^0 10^1 10^2
0.15 0.2 0.25 0.3
10^{-2}
0.1 0.2 0.3 0.4
-

Figure E.3: FT

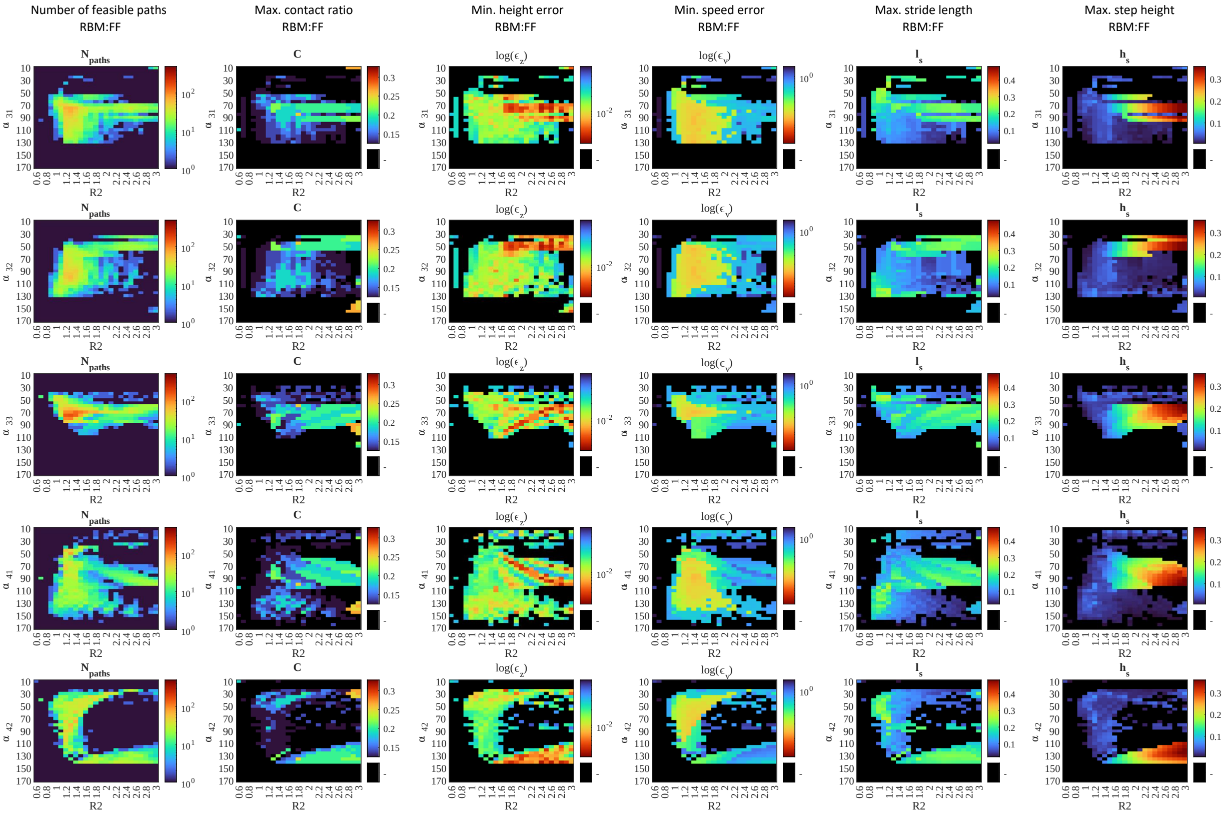

Number of feasible paths
RBM:FF
Max. contact ratio
RBM:FF
Min. height error
RBM:FF
Min. speed error
RBM:FF
Max. stride length
RBM:FF
Max. step height
RBM:FF
N paths
C
log(ϵz)
log(ϵv)
ls
hs
α 31
α 32
α 33
α 41
α 42
10
30
50
70
90
110
130
150
170
0.6
0.8
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3
R2
10^0
10^1
10^2
0.15
0.2
0.25
0.3
10^-2
10^0
0.1
0.2
0.3
0.4
-

Figure E.4: FF

E.2 Design case: mobile robot

Besides the given active surface application, the proposed constant-height walking locomotion can be utilized in other applications such as the aforementioned mobile robots. Each distinct application has its own criteria for which the geometry can be optimized.

For example, one might imagine a robot that has to navigate rough terrain with large vertical obstacles. For such a robot the speed and height error would be less important, but one would prefer a large stride length with enough step height to overcome these obstacles. Therefore, a step height criterion is defined. As shown in Figure E.5, there are two factors that can limit the step height of a robot. Firstly, the step height is limited by the height of the output path h_p: a lower path height limits the robots capability to place its foot on an elevated platform. To avoid slender skewed paths with little difference in height, this path height is defined as the distance between the highest point of the path above the propulsion phase, and height of the locomotion plane Z_{LP}. A second factor limiting the step height is the distance between the locomotion plane and the fixed body of the robot $h_b = |Z_{LP}|$. If this distance is the limiting factor, the robot body can get stuck behind obstacles sticking out of the surface, regardless of the path height. Therefore, the overall step height is determined as the minimum of these two factors:

$$h_s = \text{min}\left(h_p\right) \tag{E.1}$$

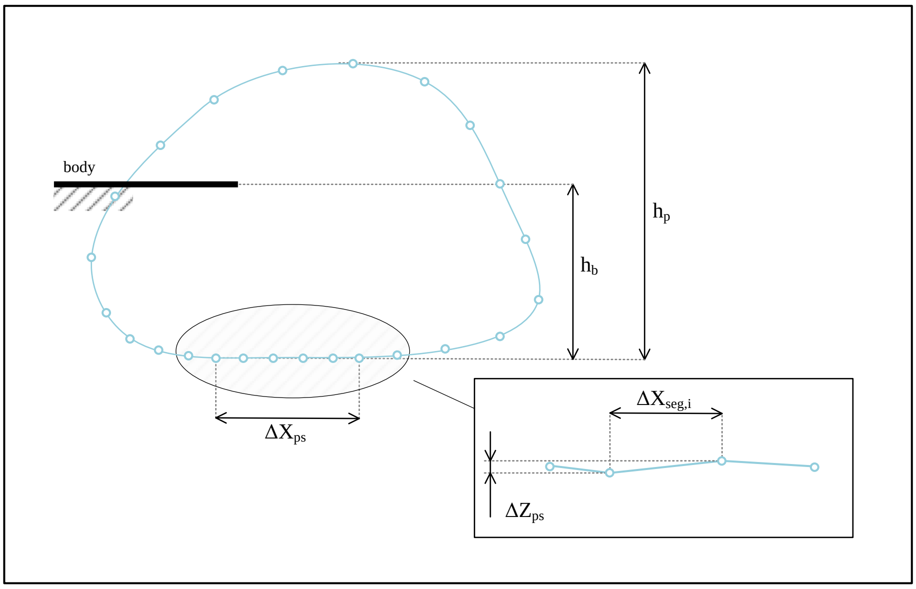


Figure E.5: Determination of the path height and height of the fixed body, which are used to determine the step height of the mechanism.

Appendix F

Auxiliary Parameters

While the parametric study results in all necessary geometric parameters to determine the output path of the end effector, multiple other geometric parameters exist that do not necessarily influence the kinematics of the mechanism, but do affect other factors. Furthermore, these allowed values of these parameters are bound to avoid parts of the mechanism from colliding with the locomotion plane. In MATLAB script `maxAuxDimensions.m`, the boundary values are calculated in as follows:

Maximum crease lengths

Firstly, the maximum lengths L_{3-5} and L_{4-9} of in-risk crease lines c_{3-5} and c_{4-9} is calculated. In order to do this, the vectors representing these crease lines, $\mathbf{v}_{3-5}$ and $\mathbf{v}_{4-9}$ are calculated for the entire input range by using a series of vector calculations, similar to the original calculations used to compute P_{EE}:

$$\mathbf{N}_A = \mathbf{R}(\mathbf{v}_{3-2}, \rho_{3-2})\mathbf{N}_Z \tag{F.1}$$

$$\mathbf{v}_{3-4} = \mathbf{R}(\mathbf{N}_A, \alpha_{3,A})\mathbf{v}_{3-2} \tag{F.2}$$

$$\mathbf{N}_D = \mathbf{R}(\mathbf{v}_{3-4}, \rho_{3-4})\mathbf{N}_A \tag{F.3}$$

$$\mathbf{v}_{3-5} = \mathbf{R}(\mathbf{N}_D, \alpha_{3,D})\mathbf{v}_{3-4} \tag{F.4}$$

$$\mathbf{v}_{4-9} = -\mathbf{R}(\mathbf{N}_D, -\alpha_{4,D})\mathbf{v}_{3-4} \tag{F.5}$$

Where $\mathbf{N_i}$ represents the normal vector of the corresponding facet F_i, and rotation matrix $\mathbf{R}(\mathbf{v}, \theta)$ represents a rotation of θ degrees around vector $\mathbf{v}$.

Subsequently, the position of vertices v_5 and v_9 are determined at the points in time their neighbouring crease lines are the steepest. Since in the script, the crease lines with no determined length are represented with a unit vector (i.e., their length is equal to the reference length: $v = R_A$) , the maximum length of each crease line can be calculated by dividing the height of the locomotion plane (Z_{LP}) through the Z-coordinates of these vertices, Z_5 and Z_9:

$$L_{i-j,max} = \frac{Z_{LP}}{Z_{j,max}} \tag{F.6}$$

Maximum facet dimensions

The maximum dimensions of the facets bordering these critical crease lines can be determined by calculating the angle between intersection of this facet with the locomotion plane, at the points in time where their neighbouring crease lines are at its steepest angles.

For instance, facet F_D is neighboring both critical crease lines c_{3-5} and c_{4-9}. Therefore, two separate angles of incidents must be calculated at two separate moments in time: $t_{3-5,max}$ and $t_{4-9,max}$. This is then used to calculate the maximum allowed angle between the outer border of the facet, and its neighbouring crease lines: $\beta_{5,D,max}$ and $\beta_{9,D,max}$. A similar process is used for angles $\beta_{5,E}$, $\beta_{4,D}$, $\beta_{4,C}$, and $\beta_{3,C}$. Note that the angle on facet F_C next to the end effector is also calculated. Accordingly, this angle is calculated according to the times point where v_{4-8} is at its steepest.

The maximum angle of arbitrary facet F_ϕ at its neighboring crease line c_{i-j} is calculated as follows (see Figure **??**):

$$\beta_{j,\Phi,max} = \cos^{-1}\left(\frac{\mathbf{n}_\Phi \cdot \mathbf{c}_{i-j}}{\|\mathbf{n}_\Phi \cdot \mathbf{c}_{i-j}\|}\right) \tag{F.7}$$

where $\mathbf{n}_\Phi$ is the intersection between facet F_Φ and the locomotion plane:

$$\mathbf{n}_\Phi = \mathbf{N}_\Phi \times \mathbf{N}_{LP} \tag{F.8}$$

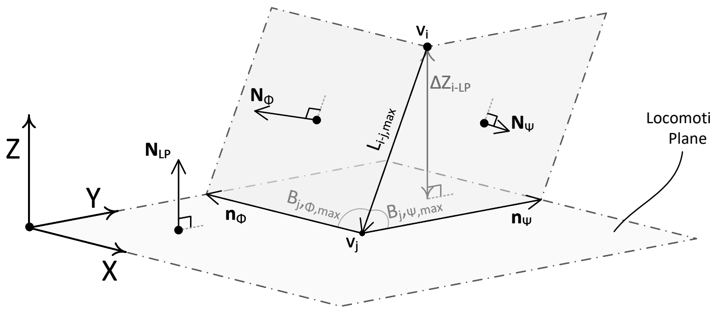


Figure F.1: The vectors used to calculate the maximum angle between facet boundaries

Selection of auxiliary parameters

The exact values of these parameters - and shapes of the facets - can be chosen according to the desired application. Furthermore, These parameters affect non-kinematics related characteristics of the mechanism such as:

- The size of the mechanism
- Facet stiffness, which in turn affects the load carrying capabilities
- Mass-related properties, such as the total mass of the facets, and the center of mass for each facet, and the accompanying moment of inertia. These can affect the properties of the system regarding multi-body dynamics, as well as the deformation in the crease lines due to sagging. These mass properties, as well as the stiffness could be further manipulated by adjusting the material, such as applying holes in the facet at places the mass or stiffness could be decreased.
- Other dynamic properties of the system such as the eigenfrequency, which can be valuable in the application of high-speed active surfaces.

Appendix G

Experimental Validation

This appendix describes the fabrication of the prototype, the test setup, and the image processing calculating the position of the end effector.

G.1 Fabrication of the prototype

As stated in the main paper, the prototype is fabricated using a technique as presented by Melancon et al. [8]. The steps are as follows, as displayed in Fig. G.1:

1. The crease pattern is laser cut onto two sheets of 400 g paper (a)
2. A layer of double-sided adhesive tape is placed between the two sheets, creating a sandwich structure with compliant inner layer (b & c)
3. The crease pattern is cut out and refined. The tape is cut away from the vertices; An incision is made in the crease lines to decrease the stiffness (d & e)
4. In order to be able to detect the position of the end-effector relative to the origin, two green markers are added which can be detected during the image processing (f)
5. The secondary dimensions are applied by cutting away excessive material (g).

The best prototype is achieved with a crease width of 0.8 mm. With a larger crease width, unwanted degrees of freedom are introduced, such as rotation orthogonal to the crease line, and displacement normal to the facets bordering the crease. For smaller crease width, an increased stiffness in the crease lines causes excessive bending in the facets. Furthermore, facet thickness plays a much larger role for small crease widths, causing the neighbouring facets to collide when the fold angle gets too high.

G.2 Test setup

For experimentation, the fixed facet of the crease pattern is clamped onto a platform. In order to determine the output path of the end effector, its position is fixed for 16 distinct moments in the gait cycle. The input angles ρ_{1-3} and ρ_{2-3} of these 16 distinct moments are fixed using 3D-printed clips (see Fig. G.2a, right), each with an angle corresponding to the input profile at that time step, as shown in Table **??**.

At each of these time steps, a picture is with a camera positioned on the negative Y-axis. Using the created image processing code `imageProcessing.m` in MATLAB, the position of the end effector relative to the origin is calculated. Firstly, the image is calibrated by measuring length R_A such that the measured distances can be accurately converted. Subsequently, for every time step, the green dots are isolated and transformed, and their centroids are calculated. These are then used to measure the distance between the dots, which is translated into the end-effector coordinates.

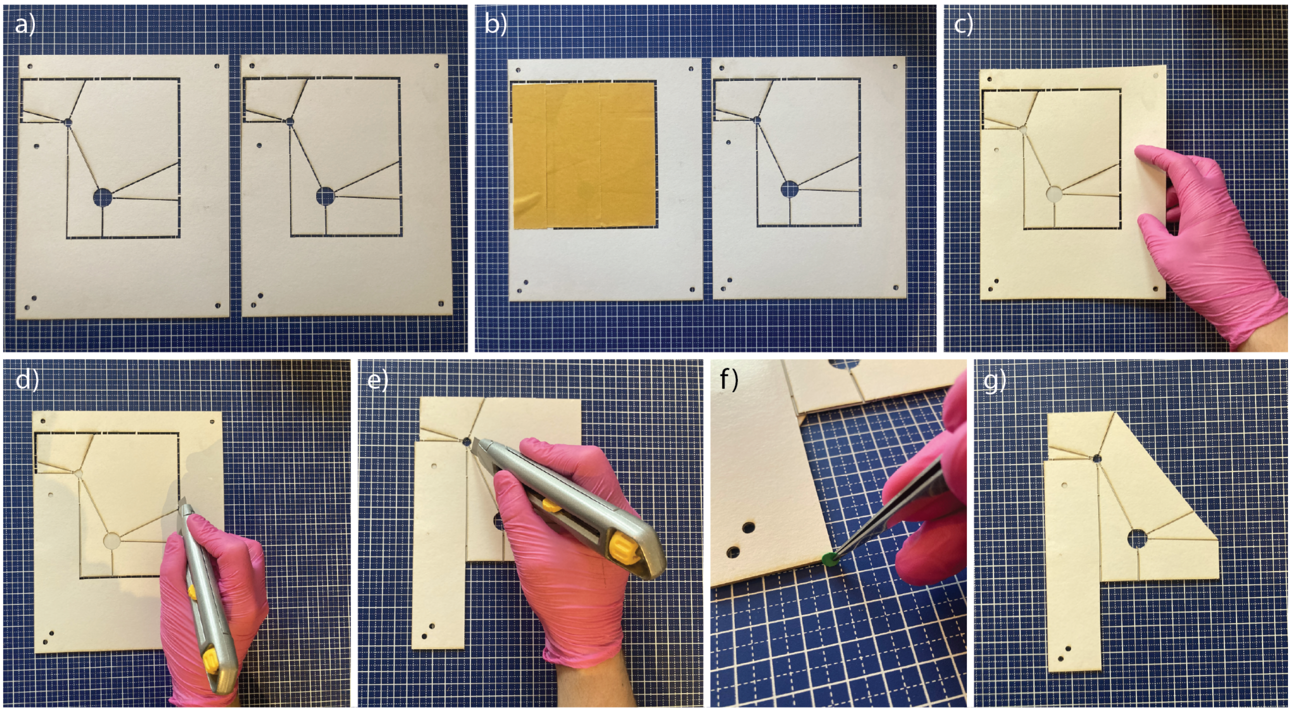


Figure G.1: The steps taken in the fabrication of the prototype.

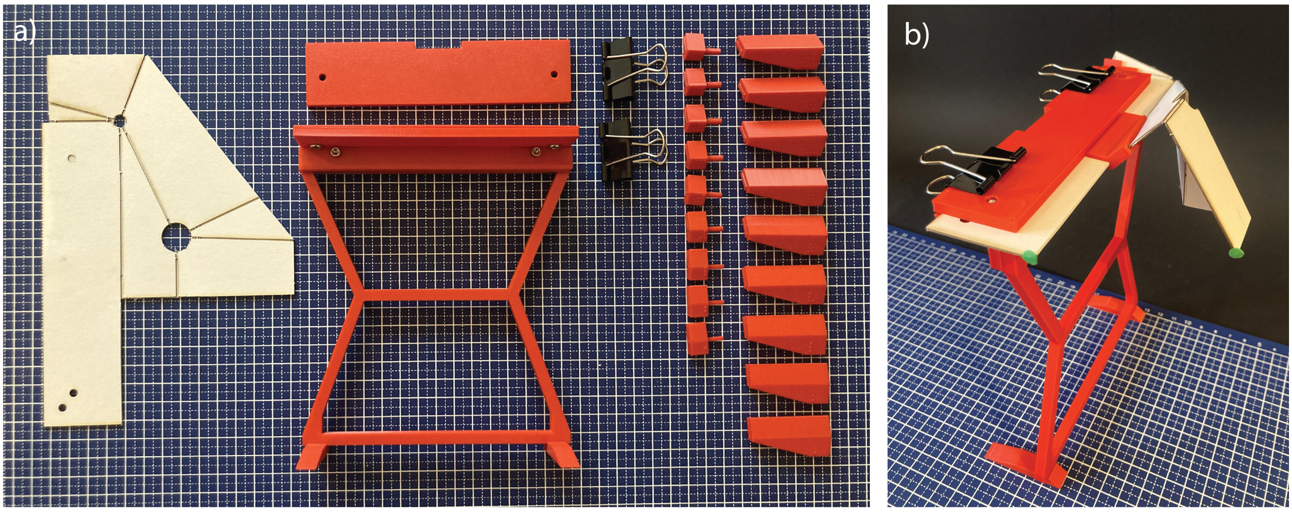


Figure G.2: a) An overview of all the materials used in the test setup, including the crease pattern itself, the 3D-printed platform on which the mechanism is elevated, and the clips used for input ρ_{1-3} and ρ_{2-3}. b) The test setup used for the experiment.

Bibliography

[1] D. Rus and M. T. Tolley, “Design, fabrication and control of origami robots,” *Nature Reviews Materials*, vol. 3, pp. 101–112, 2018.

[2] P. S. Gollnick, S. P. Magleby, and L. L. Howell, “An introduction to multilayer lamina emergent mechanisms,” *Journal of Mechanical Design, Transactions of the ASME*, vol. 133, no. 8, pp. 1–11, 2011.

[3] D. Piker, “No Title,” 0.

[4] J. L. Silverberg, J. H. Na, A. A. Evans, B. Liu, T. C. Hull, C. D. Santangelo, R. J. Lang, R. C. Hayward, and I. Cohen, “Origami structures with a critical transition to bistability arising from hidden degrees of freedom,” *Nature Materials*, vol. 14, no. 4, pp. 389–393, 2015.

[5] L. Zimmermann, T. Stankovic, and K. Shea, “A Computational Design Synthesis Method for the Generation of Rigid Origami Crease Patterns,” *Journal of Mechanisms and Robotics*, vol. 14, 6 2022.

[6] A. Walker and T. Stankovic, “Algorithmic design of origami mechanisms and tessellations,” *Communications Materials*, vol. 3, no. 1, pp. 1–8, 2022.

[7] T. Moller, “A Fast Triangle-Triangle Intersection Test,” *Journal of Graphics Tools*, vol. 2, no. 2, pp. 25–30, 1997.

[8] D. Melancon, B. Gorissen, C. J. García-Mora, C. Hoberman, and K. Bertoldi, “Multistable inflatable origami structures at the metre scale,” *Nature*, vol. 592, no. 7855, pp. 545–550, 2021.