# Design Tools for Foldable Structures with Application of Fuzzy Logic

Madalina N. Wierzbicki
and Clarence W. de Silva

# Design Tools for Foldable Structures with Application of Fuzzy Logic

Madalina N. Wierzbicki and Clarence W. de Silva

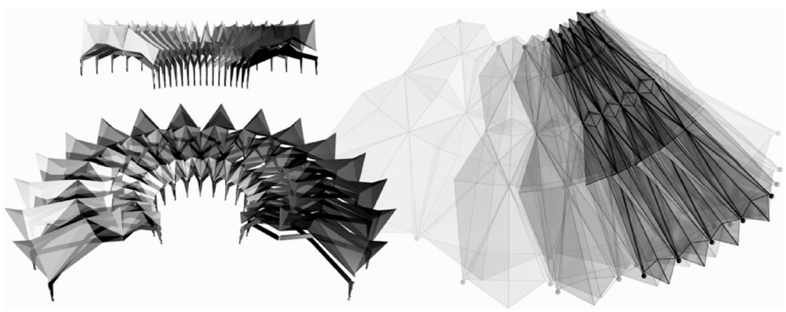
Rigidly foldable shells offer tremendous potential for developing kinetic architectural structures. However, the added element of motion poses new design challenges. Initially, sketchy shell geometry is constructed to reflect the intended form. Further steps involve assuring an error free folding within a range that satisfies desired functional requirements. The kinematics of a parallel topology of the shell's geometry is difficult to express algorithmically what prevents from developing of automated adjustment tools based on computational methods. The geometry can be adjusted manually based on intuitive observations; however the process is tedious, time consuming and unpredictable. This paper develops automated adjustment tools based on the intuitive approach of a human designer. The study applies the fuzzy logic formalism as a computational interface between human approach and structured adjustments to the geometry. The advantages of fuzzy logic stem from its natural ability to represent human knowledge and effectiveness in reconciling ambiguities, uncertainties and redundancies that the intuitive human approach brings along. The development steps of fuzzy logic based algorithm are presented. Performed evaluation tests and the results are discussed.

## 1. THE DEVELOPMENT AND APPLICATIONS OF FOLDABLE STRUCTURES

The idea of configurable, kinetic elements in architecture is as old as architecture itself. Doors, windows, blinds, gates, draw bridges and other simple in concept devices were known for ages. Recent decades brought large retractable roofs over stadiums and stages, elaborate partition systems in conference centers, configurable floor installations in concert halls, anti sway devices in high rise towers and automatically adjustable columns that compensate for ground settling. However, these kinetic elements perform mostly a limited, localized function of controlling egress and adjusting some aspects of exterior shell or the structure. The concept of a 'foldable' building or reconfigurable on demand functional layout is not here quite yet despite the progress in design tools and material technologies.

Foldable structures may offer features that are beneficial while responding to the extended set of contemporary requirements. These requirements call for better safety when faced with natural and man induced disasters. They encourage meaningful environmental responsibility. Foremost, they define the functional and comfort expectations of societies that are becoming more dynamic and complex. Kinetic structures prefabricated as modular components can be assembled on site with lesser environmental impact than buildings constructed by traditional methods. Furthermore, the inherent ease of disassembly and reuse of modular kinetic components facilitates lifecycles based on adjustment and relocation rather than demolition and new construction. Functionally, kinetic systems have the capability to play an important role in forming reactive environments, spaces that can intelligently respond to changing conditions like occupancy dynamics, environmental conditions and emergencies.

Undoubtedly, an added element of kinematics will introduce new challenges for designers. This paper examines the process of developing a sample folding structure illustrated in Figure 1 and focuses on difficulties encountered while testing the folding performance of the geometry. In particular, physical demands of folding require eliminating of occasional



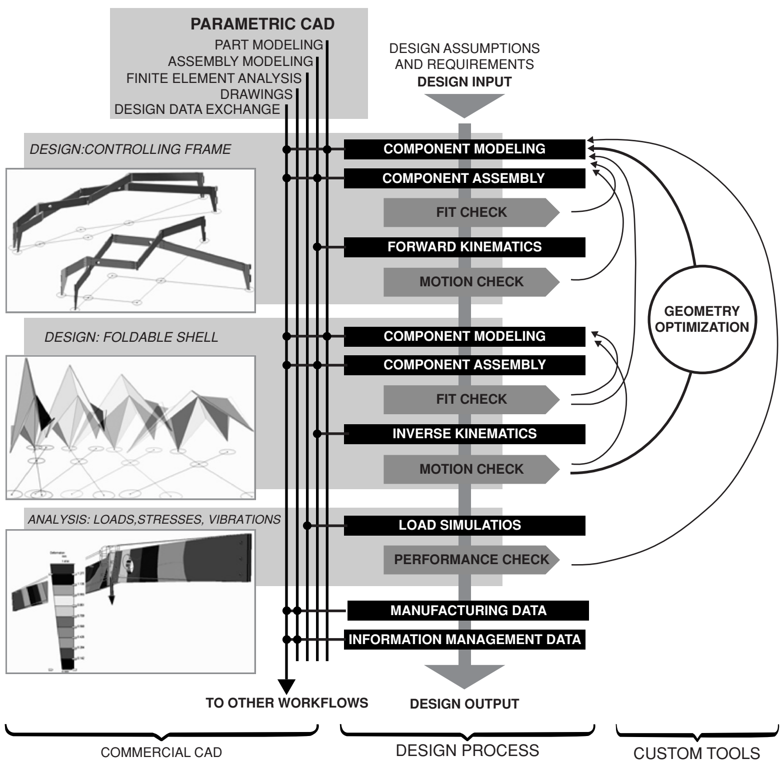
► Figure 1. The concept of a folding structure.

collisions and intersections between the elements of the shell that may occur within the range of motion. Folding errors can be corrected by adjusting the geometry manually. However, such approach is tedious, time consuming and unpredictable. The paper discusses the need for efficient design tools, explains the limitations of formulaic algorithms and evaluates approach that allows building automated design tools based on intuitive observations designers gain during motion simulation studies.

## 2. DESIGN WORKFLOW

The scope of a traditional architectural workflow is already undergoing expansion as motion and actuation elements migrate gradually into building designs. The design of kinetic architectural structures will expose architects to unique demands of modeling of the translation geometry and simulation of the motion range. The design tools that perform similar tasks are widely available in mechanical engineering. In particular, parametric assembly environments offer the ease and flexibility of both forward and inverse kinematics for modeling of complex kinetic designs.

Integrated CAD environments provide comprehensive design toolkits that can be effectively utilized within various project delivery workflows. Figure 2 illustrates a fairly typical CAD toolset interleaved with a




◀ Figure 2. Design workflow and design tools.

workflow aimed at designing folding structures. The workflow is based on logical process iterations that loop through design and verification/validation steps until the transition from design input to design output is complete. The initial concept undergoes a series of modifications as it progresses through development stages. The first step is to outline the intended form with mesh facets constrained to the controlling frame. Typically, the intended form defines the unfolded state of the structure which represents the target functional, layout and aesthetic requirements to be fulfilled. The next step determines the adjustability range of the folding shell. Adjustability is one of the defining design criteria that set it apart from traditional building structures. It is also a parameter that needs to be validated early in the development stages as it drives a number of design detailing dependencies like element sizing, folded state envelope, actuation and articulation. Such validation focuses on the kinematic characteristics of the geometry.

The key feature that helps with tackling the complexities of foldable structures is the flexibility of switching between bottom-up and top-down methods while developing the design. The initial concept of the controlling grid and the folding frame can easily be put together while starting with components, progressing to the assembly level and then employing the capacity of forward kinematics modeling. This is a straightforward bottom-up design strategy. The development of a foldable shell requires a different approach. The kinetic complexity of foldable meshes makes algorithmic descriptions tedious to define. However, a loosely modeled set of facets that follows locally applicable geometric constraints can be easily appended to the already defined controlling grid and verified by means of inverse kinematics.

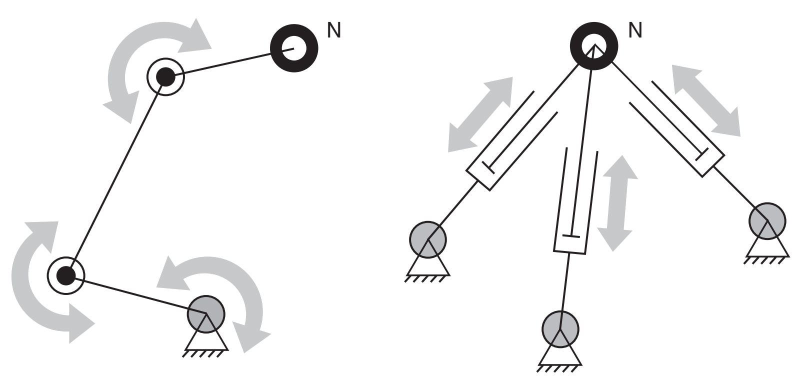
The initial draft can be further refined through adjustments to the geometry. One of the key challenges is eliminating of any errors while folding. The efficiency of this process relies on the seamless integration of solid parametric modeler with parametric assembly modeler. Once the components can be modified directly within the environment of kinematic simulation, the development process focuses on optimization of the form without the cumbersome burden of switching between different design environments. However, CAD tools have limited use when determining what adjustments need to be applied. The reason, as explained in more detail in Paragraph 3, is difficulty in developing algorithmic relations between the adjustments and their effects for shells that are comprised of many facets. The decisions concerning the adjustments have to be made by the designer outside of the boundaries of the CAD environment. This design step marked as the 'GEOMETRY OPTIMIZATION' is an important factor affecting the overall efficiency. If much slower or less reliable than the rest of the CAD environment, it becomes the performance bottleneck of the whole process. Consequently, if not resolved, it may restrict the integration of new solutions with mainstream practices.

The diagram depicted in Figure 2 highlights the design step that falls out of the boundaries of typical CAD environments. This paper discusses in particular the challenges in automating this step using formulaic methods and presents an alternative approach.

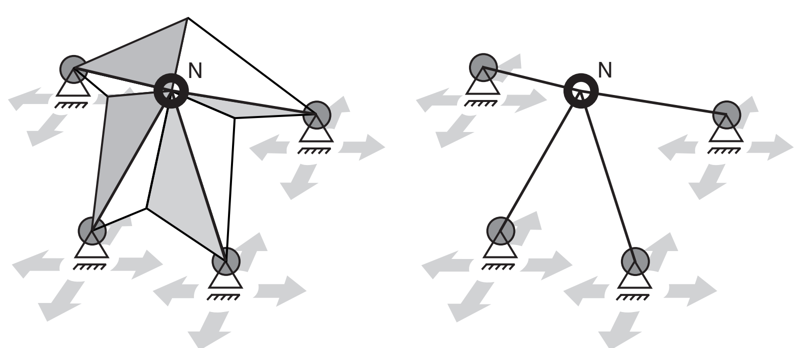
## 3. OPTIMIZATION CHALLENGES

Computational challenges encountered in designing industrial robots provide good context for understanding folding of shells. Two basic geometrical topologies of robots, serial [1] and parallel [2], are illustrated in Figure 3. In the serial configuration, linkages form a chain where only a single linkage attaches to the end node (N). In the parallel configuration, multiple linkages attach simultaneously to the end node (N).

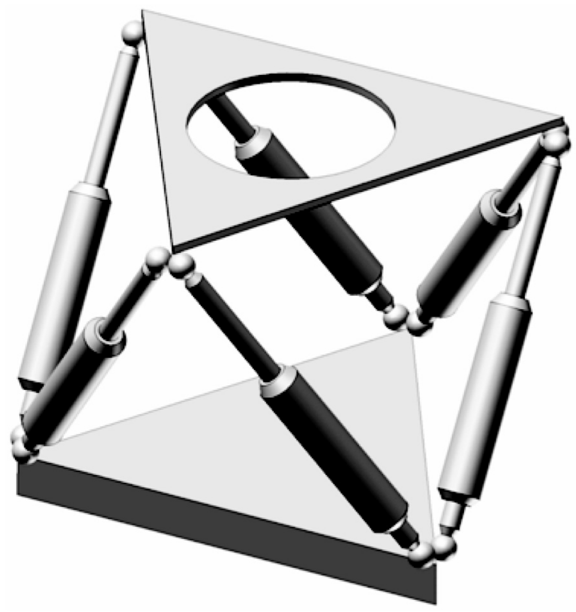
The topology of a foldable shell is analogous to that of parallel robots, as shown in Figure 4, since multiple supporting ridges attach simultaneously to the supported vertex (N). Such parallel geometries are difficult to calculate in terms of forward kinematics. If all the positions of the controlled ends of the ridges are given, expressing the position of the supported vertex (N) as a set of formulas can be challenging even for relatively simple geometries.




◀ Figure 3. Left: Kinematic diagram of a serial robot. Right: Kinematic diagram of a parallel robot.




◀ Figure 4. Left: Kinematic diagram of a foldable mesh. Right: Parallel topology of the mesh ridges.



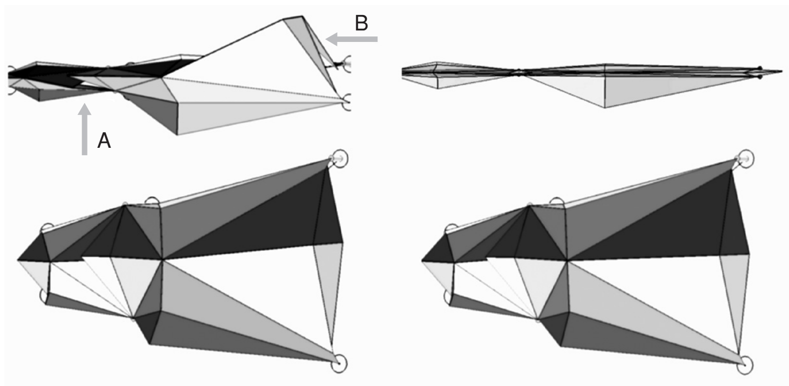
► Figure 5. The topology of Gough platform.

For example, an industrially successful implementation of a parallel robot known as Gough platform, Figure 5, was first documented by V. E. Gough in 1956 [3].

However, the research on forward kinematics of this six strut parallel device has been progressing slowly. The number of possible theoretical forward kinematics solutions was identified in 1991 [4] while only in 1998 all possible poses were calculated [5]. Regardless of this progress, an algorithmic model has not been developed yet.

In contrast, reverse kinematics is straightforward for parallel geometries as end nodes' locations explicitly define the sizing and position of linking ridges. As a result, three-dimensional meshes are easy to use for approximating a desired spatial form. Motion simulation is a practical and readily available method of evaluating their kinematic performance in terms of range of motion and the final degree of folding. In most cases, the initial shell will fall short of the desired motion performance requirements due to interferences between facets or singularities that occur while folding. Consequently, the initial geometry needs to be adjusted to improve the folding range. Any systematic approach to solve this problem has to be based on forward kinematics methods because the process of eliminating of folding errors relies on localized adjustments to ridge lengths while the nodes' positions are the outcomes of these adjustments. Generally, such adjustments are quite subtle as illustrated in Figure 6.

Despite obvious differences while in the folded state, as shown in top row of Figure 6, where the non-optimized shell demonstrates an interference




◀ Figure 6. Left: Non-optimized geometry. Right: Optimized geometry.

(A) and facets folding onto each other (B), it is virtually impossible to tell the geometries apart when unfolded, as shown in the bottom row. Obviously, in a constructed physical object such errors would result in failure and damage. Therefore any motion errors must be completely eliminated before the structure can be build and subjected to folding. Usually, even a sketchy, initial concept of a shell will be able to fold to a certain, minimal degree before errors occur. Once the geometry is corrected to eliminate these errors, the shell can fold further. Therefore the process of eliminating of the folding errors is synonymous with increasing of the motion range.

The task of increasing of the folding range of a system modeled as a kinematic geometrical construct can be expressed as an optimization problem [6] given by Eqn (1). The design variables are the shell's mesh ridge lengths expressed as vector $\boldsymbol{x}$, upper and lower bounds ($\boldsymbol{x}_{lb}$, $\boldsymbol{x}_{ub}$) of ridge lengths are dictated by the overall geometry size and shape, equality constraints are the nodes of the controlling frame expressed as vector $\boldsymbol{h}$, inequality constraints are the limits of the desired geometry envelope expressed as vector $\boldsymbol{g}$, objective ($J$) is the desired degree of folding.

$$\begin{gathered}\text{find } \boldsymbol{x} \text{ that minimizes } J(\boldsymbol{x}) \\ \text{subject to } \boldsymbol{g}(\boldsymbol{x}) \leq 0,\ \boldsymbol{h}(\boldsymbol{x}) = 0 \text{ and } \boldsymbol{x}_{lb} \leq \mathbf{x} \leq \boldsymbol{x}_{ub}\end{gathered} \tag{1}$$

In a general case, the objective would be formulated as the target of a Multidisciplinary Design Optimization (MDO) [7] and expressed as a weighted result of many requirements like optimum folding, optimum structural strength, optimum actuation and optimum cost. These contradictory requirements represent various aspects of physical and functional performance and are normally negotiated by a designer through MDO. The smaller the shape a structure can be folded into when not in use, the better. The perfect degree of folding is a complete flat fold. Under realistic conditions such fold can never be executed because of material thicknesses and assembly offsets. Perfect structural strength calls for high level of structural redundancy and large safety margins which result in increased material sections thus further reducing the degree of folding. The ideal cost implies budget economy what may preclude using of some high strength materials to reduce part sections and increase the degree of folding. The ideal actuation implies fast and reliable motion however it may require large size components and large separation zones from the end travel points to avoid singularities and the resulting indeterminate directions and infinite forces. Such oversized components and separation zones increase the size of interfacing structural elements and further reduce the attainable degree of folding. The complexities of seemingly irreconcilable dependencies are sometimes termed as 'wicked' problems [8]. The compromise values of these requirements as negotiated through MDO became optimum targets that can be attained under the initially assigned levels of importance. However, for the purpose of clarity and focus on issues pertinent to the kinematics within the working range of parallel geometries, the objective can be expressed solely as the degree of folding while structural strength, actuation and other requirements are culled into a reasonable, discrete and non null value that becomes the target for the degree of folding.

Though the desired degree of folding can be expressed as an optimization target and many optimization methods are available, they can not be utilized for solving the present problem because of practical impossibility of developing algorithmic, forward kinematics models for parallel geometries. Especially in view of the fact that even simple arrangements of 3d meshes tend to be much more complex, from kinematics point, then a six linkage topology of the Gough platform. For solving rigidly foldable shells and, for that matter, any complex parallel geometry, forward kinematics formalism does not offer, at present, any usable methods. Therefore formulaic approaches as described in the study of manufacturing of paper clips and other folded structures [9] can not be applied to folding shells.

The only immediately available recourse is a manual approach based on trial and error. Iterations of adjustments to lengths of ridges and folding simulations can be performed to seek an improvement of the degree of folding. The process is tedious and time consuming even for relatively simple geometries.

The issue of developing manageable tools for optimization of kinematic performance may be the deciding factor in considering successful commercialization of kinetic structures based on rigidly foldable shells as mainstream design workflows traditionally depend on methods that can assure a certain degree of predictability, productivity and ease of application.

## 4. FUZZY LOGIC BASED OPTIMIZATION

For a foldable structure, manual adjustments of shell elements are based on readily observable local dependencies between ridge length changes and overall kinematic behavior changes. In general terms, the manual methods rely on accumulated experience and learned knowledge about relationships between element sizing, proportions and exception occurrences. Intuitive and experience-based adjustments are applied in gradual steps to the initial geometry until the desired degree of folding is achieved. Obvious disadvantage of the manual method is unpredictability which is the result of the underlying subjective, trial and error approach. Also apparent is the difficulty to automate this process since missing is a clearly defined algorithm that links the performance (the degree of folding) with the sizing of the shell geometry.

The study presented in this paper proposes to use the formalism of fuzzy logic to harness the potential of human observations and the resulting intuitive and knowledge-based conclusions for the purpose of developing algorithms that can perform reliably, predictably and efficiently in structured workflows. The goal is to automate the geometry related decision making while designing folding structures. Fuzzy logic has been intended as a tool capable of interpreting human knowledge within the context of a specific application and reasoning with that knowledge to make useful inferences or actions [10]. It provides a systemic framework for using intuitive, often imprecise and conflicting instructions based on human knowledge to build robust and reliable algorithms for system automation.

In the present work, a set of typical steps one would perform while adjusting a real life prototype of a folding structure was examined. Such exercise yields numerous clues and hints about the kinematic behavior of the geometry and possible ways of improving the degree of folding. Based on readily observable occurrences like collisions or slopes and intuitive remedial actions like shortening or lengthening of linkages, the knowledge acquisition process can capture 'cause and effect' dependencies in the shell geometry which can be formulated as 'If-Then' statements. For example, a rule may take the form: if a set of neighboring facets would fold onto each other while the shell folded very little then increase the appropriate dimensions of the bottom facet and reduce the appropriate dimensions of the top facet. Though simple and localized in their scope, such statements are sufficient to build the so called Knowledge Base which can interface with fuzzy decision making (or, inference engine) to identify problem conditions and infer appropriate corrective actions. Figure 7 shows examples of statements as compiled for the Knowledge Base.

The advantage of the fuzzy logic formalism lies in its effectiveness when dealing with human like approach which, while intuitive and easy to understand, usually brings along ambiguities, contradictions, uncertainties

▶ Figure 7. Examples of If-Then statements.

**IF** (*Degree-of-Folding* **IS** *Unfolded*) **AND** (*Relative-Crown-Node-Height* **IS** *Lower*)
**THEN**
(*Lengthen-Subject-Linkage* **IS** *Significantly*)

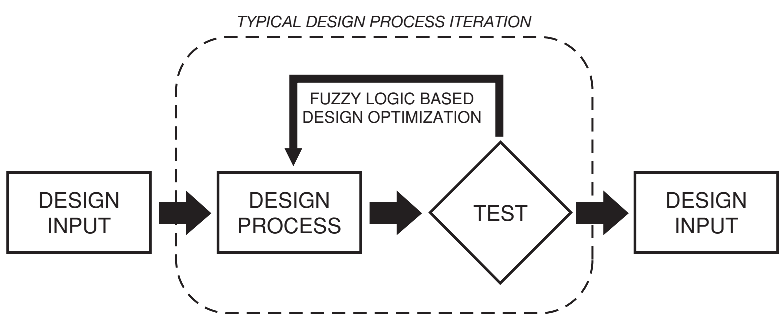
**IF** (*Degree-of-Folding* **IS** *Unfolded*) **AND** (*Relative-Crown-Node-Height* **IS** *Equal*)
**THEN**
(*Lengthen-Subject-Linkage* **IS** *Significantly*) **AND** (*Shorten-Sibling-Linkage* **IS** *Significantly*)

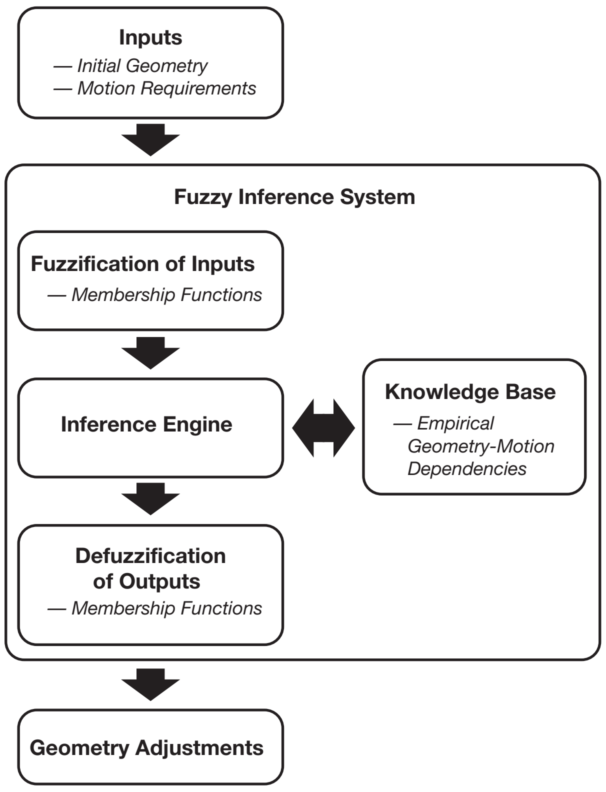
and redundancies. Experiential clues are local in nature as they relate to the immediate circumstances of an observed error. Fuzzy Logic allows inferring of geometry adjustments based on these local and loosely defined indications. More importantly, it allows forming algorithms that attempt to reconcile iterations of such adjustments within the context of the performance of the complete shell thus negotiating contradictions and overlaps between locally made decisions.

A design process delivers a solution to a set of initial requirements which outline the desired end product. What happens in-between the initiation and completion of a design process is a subject of much research and rhetoric [11]. However, initial to final point approach provides an external, high level view that allows to treat the design process in a simplified manner as an input-output block which has been termed in related research as the 'basic design procedure' [12]. The solution, typically, is a result of numerous iterations through design modification, testing and review stages. Research on design methodology uses a variety of similar, in meaning, terms to describe these stages like 'analytical, creative and executive phases' and consistently defines them as 'the main phases of design' [13]. A Fuzzy Inference System (FIS) built around a set of intuitive and knowledge-based directives is intended to govern the modification loop in the design workflow, as shown in Figure 8.

The purpose of implementing Fuzzy Logic for the design of kinetic structures is to avoid the prohibitively difficult forward kinematics modeling of complex parallel kinematic assemblies. Instead, designers would devise a

▶ Figure 8. Block diagram of the design process.





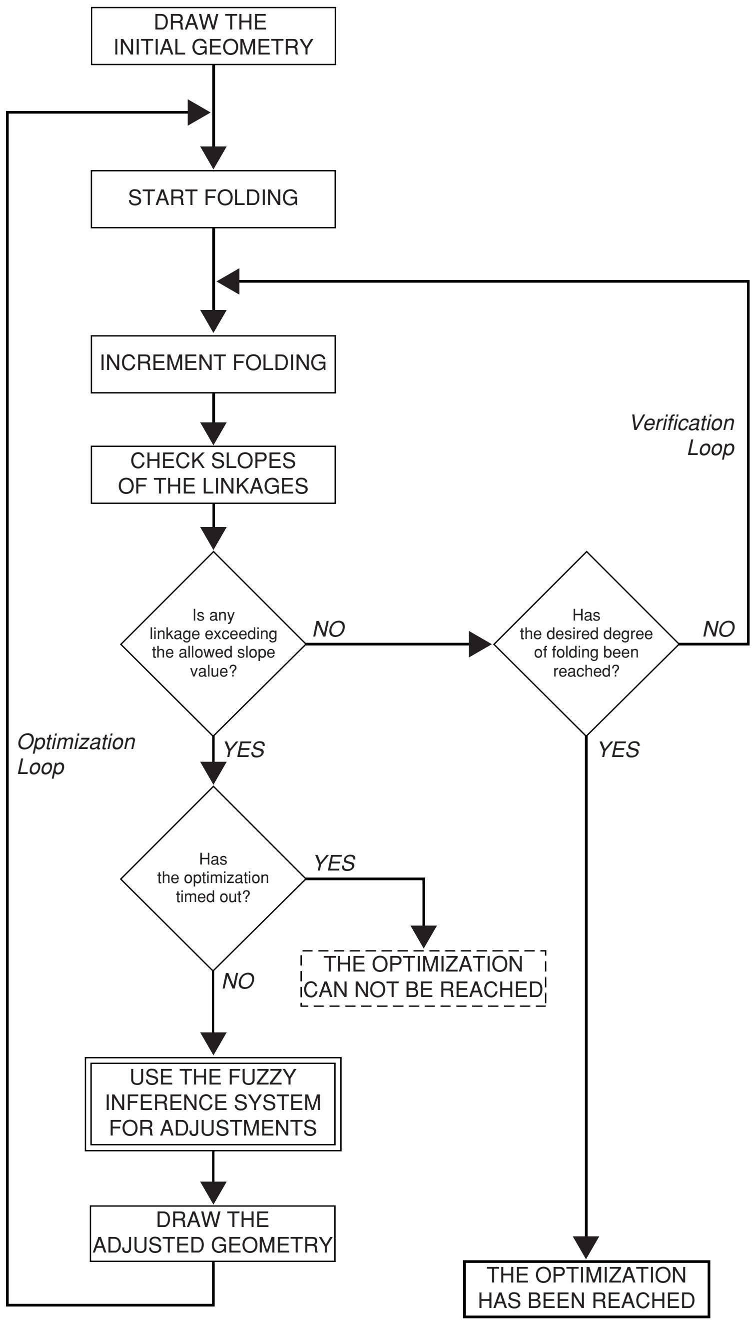

◀ Figure 9. Block diagram of FIS.

sketchy geometrical topology and rely on the optimization process implementing a FIS to adjust the linkage sizing. Figure 9 illustrates a block diagram of the conceived algorithm based on fuzzy logic. The cornerstone of such algorithm is a set of simple, intuitive, knowledge-based and empirically derived adjustment dependencies compiled in the Knowledge Base in the form of if-then statements. Fuzzification and defuzzification stages provide the means of quantitative interfacing with the crisp domain of the geometrical model. Membership functions express assignments of concrete ranges of values to qualifiers like 'significantly' and 'little' used to build statements in the Knowledge Base.

The Inference Engine uses the input data as translated through input membership functions and infers appropriate adjustments based on the if-then statements contained in the Knowledge Base. Further, the appropriate 'then' components are translated into numerical values of adjustments using the output membership functions.

A detailed, programmatic flow of the geometry adjustment tool is illustrated in Figure 10. The optimization algorithm is inserted into a typical design iteration loop where a design modification is performed and, then, followed by a qualifying test step. Depending upon the results of the test, two possible outcomes would be available: either further modification steps

▶ Figure 10. Block diagram the optimization algorithm.




are continued or the loop is terminated with an optimized solution. Hence clearly defined are two distinct process flow loops: verification (on the right side) and optimization (on the left side). A Fuzzy Inference System plays the key role in the optimization loop where it generates the necessary adjustment data. For practical programmatic reasons, a timeout test is inserted. It prevents run-away code execution if the optimization cannot be reached.

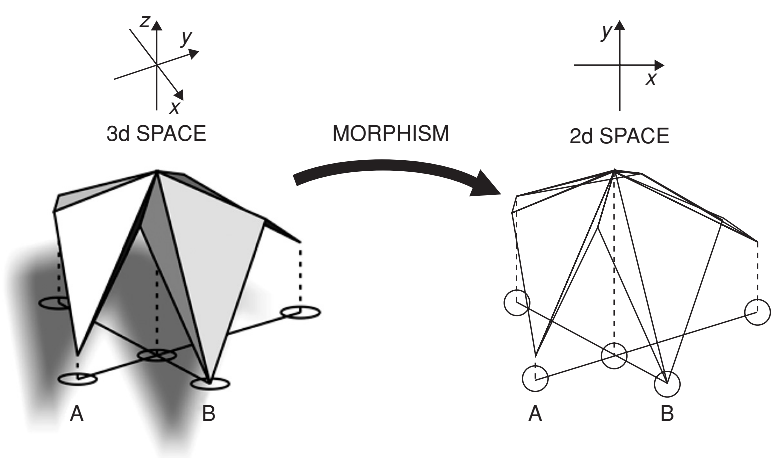
The algorithm is intended to utilize, through its FIS module, the empirically observed dependencies to infer appropriate geometry adjustments. It is set to perform all the necessary steps automatically in a progressive succession where the worst, from the folding viewpoint, areas of the shell are adjusted first.

The goal is to achieve the optimization of larger geometries by iterating through localized corrective actions on geometry fragments. This scalable nature of the scheme would potentially enable optimizing of very complex geometries.

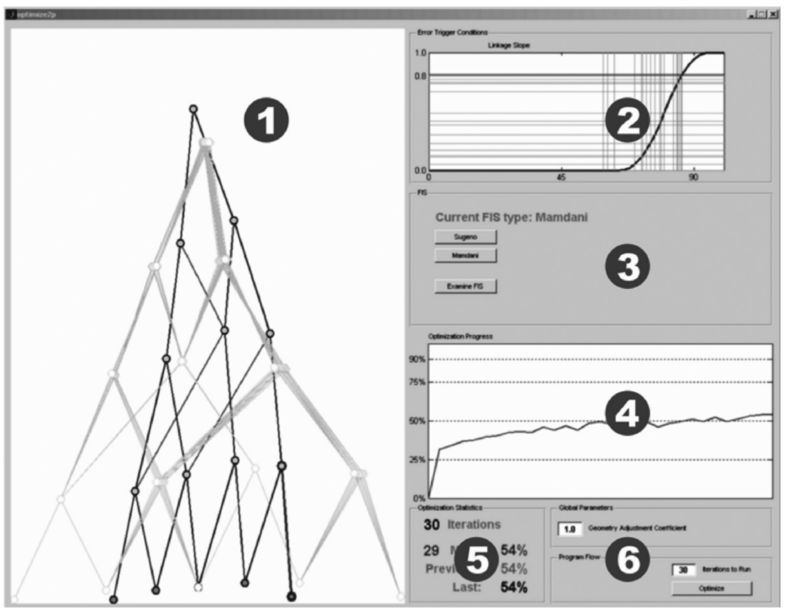
## 5. TESTING OF THE ALGORITHM

It is important for the test simulation to be able to model accurately the key kinematic effects that occur in a 3d shell that is being folded. It is also beneficial for the efficiency of the simulation to avoid modeling of features and phenomena that are not relevant for the kinematic aspects of the folding. Kinematics deals with non-real-time, non-dynamic aspects of the subject geometry. In this case, the focus is to achieve the desired motion range by adjusting of the sizing of the geometry while retaining its original topology. Issues of load bearing, vibration and actuation do not affect the degree of the folding; therefore they are not considered during this stage of the design process. Also, the general shape of the envelope as set by the initial topology is not the subject of this optimization.

While developing the simulation model, advantage has been taken of the morphism between 3D and 2D spaces, and particularly of the fact that a 2D projection of a 3D mesh is, topologically, a continuous function [14], as shown in Figure 11.




◀ Figure 11. Mapping of geometry between topological spaces.




► Figure 12. Testing setup interface.

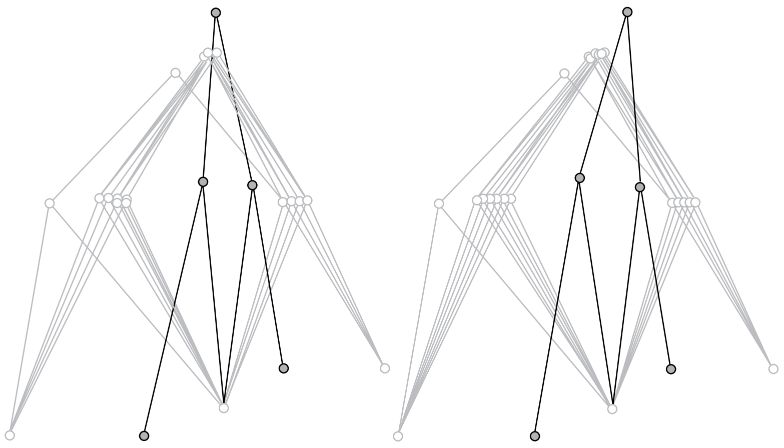
As such, it maps the parallel character of the geometry, the motion range as well as all the kinematic exceptions like interferences and singularities from one space onto the other. For instance, if the distance between A and B diminishes in 3d space, it will also be diminishing in its 2d projection. The detection of interferences and singularities (fully stretched or flat folded states) and performing the adjustments that eliminate them is the objective of the optimization algorithm. Therefore, a simulation model that employs a 2d lattice instead of 3d mesh is equally effective in validating the performance of such algorithm. The advantages of using a 2d construct for simulation include simpler programming, effective use of computer resources and the generation of a fast-executing simulation code. Such streamlined approach allows sufficient testing of the algorithm and implementation of necessary parameter adjustments within a reasonably short time.

A virtual testing experiment has been set up entirely in Matlab while utilizing its Fuzzy Logic Toolbox.

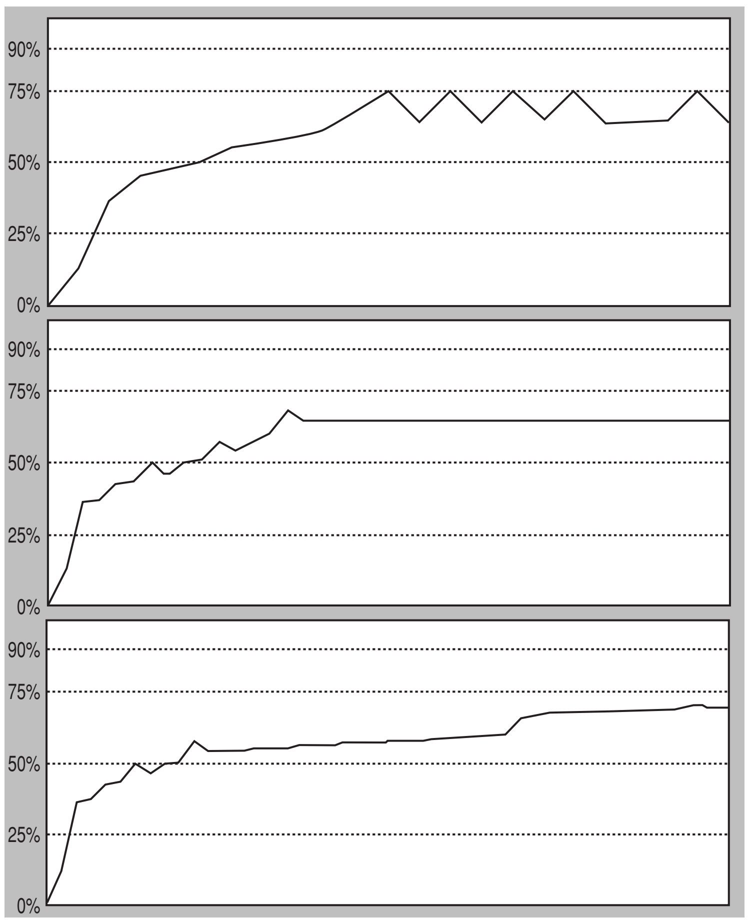
The main components of the interface, Figure 12, are:

- The lattice plot window where the folding is shown in real time (1).
- Trigger condition plot window which displays, in real time, all linkage slopes imposed over the slope membership function (2).
- FIS edit and review pane, center right (3).
- Optimization progress plot (4).
- Optimization statistics (5).
- Program parameters and controls (6).

Figure 13 illustrates sample optimazations.



◀ Figure 13. Optimization examples.




◀ Figure 14. Typical behaviors of the algorithm.

These optimizations iterated 20 times and reached, in both cases, the folding rate in excess of 78% what is a notable improvement if compared to the initial folding rate of 12%.

Figure 14 illustrates typical behaviour of the algorithm.

Different behaviors reflect changes made to the magnitude of adjustments as coded through membership functions. The bottom behavior is the desired one as it assures that the last performed iteration results in the highest degree of folding achieved thus simplifying automation of the software.

## 6. CONCLUSIONS

The goal of the study was developing of a tool that could automate adjustments to the geometry of folding shells in order to increase the degree of folding. Fuzzy logic has been utilized to build an algorithm based on very simple if-then statements derived from intuitive human observations. The algorithm is intended as an option to formulaic approach which tends to be prohibitively complex when applied to folding shells. The algorithm has been tested on lattices of various complexities. Despite its simplicity and reliance only on local dependencies instead of global, high level analysis of the geometries, the algorithm fared well in increasing the degree of folding. Sample geometries ended up folding significantly more (from initial 12% to in excess of 70%) within a reasonable amount of iterations (15 to 40). Also, the algorithm responded well to adjustments of values defining the membership functions by demonstrating a broad range of dynamic behaviors as shown in Figure 14.

Therefore the algorithm has satisfied the essential expectations of a tool: efficacy and controllability. These promising results encourage continuing the work on algorithms that implement fuzzy logic to harness the potential of simple, intuitive and experientially derived human observations. Further work will focus on performance improvements (speed and achieved degree of folding), reliability for border condition geometries and effective preservation of the initial layout intent.


## Acknowledgements

The author would like to thank Dr. Elisabeth Croft, Dr. Don H. Krug, Dr. Sherry McKay and other colleagues at the University of British Columbia for encouragement, expertise, and moral support during the novel interdisciplinary research project presented in this paper.


## References


1. Craig, J.J., *Introduction to Robotics,* Addison-Wesley, 1989.
2. Merlet, J., *Parallel Robots,* Kluwer, 2000.

3. Gough, V. E., *Automobile stability, control, and tyre performance,* Automobile Div., Inst. Mech. Eng., 1956.
4. Dietmaier, P., The Stewart–Gough platform of general geometry can have 40 real postures, in: J. Lenarcic and M. L. Hust, eds., *Advances in Robot Kinematics: Analysis and Control,* MA, Kluwer, 1998, 7–16.
5. Raghavan, M., The Stewart platform of general geometry has 40 configurations, *ASME Design and Automation Conf.,* Miami, FL, 1991, vol. 32(2), 397–402.
6. Siddal, J. N., *Optimal Engineering Design,* CRC, 1982.
7. Eschenauer, H., Koski, J. and Osyczka, A. ed, *Multicriteria Design Optimization,* Springer Verlag, 1990.
8. Rittel, H. and Webber, M., Dilemmas in a General Theory of Planning, *Policy Sciences,* 1973, vol. 4, 155–169.
9. Arkin, E., Fekete, S., Mitchell, J., An algorithmic study of manufacturing paperclips and other folded structures, *Computational Geometry,* 2003, vol. 25(1-2), 117–138.
10. Karray, O.F. and de Silva, C., *Soft Computing and Intelligent System design,* Pearson, 2004.
11. Cross, N., Forty years of design research, *Design Studies*, 2007, vol 28, 1–4.
12. Cross, N., *Developments in Design Methodology,* John Wiley & Sons, Chichester, 1984.
13. Cross, N., *Engineering Design Methods: Strategies for Product Design,* John Wiley & Sons, Chichester, 1994.
14. Artin, M., *Algebra,* Prentice Hall, 1991.

Madalina N. Wierzbicki and Clarence W. de Silva
Industrial Automation Laboratory
Department of Mechanical Engineering
The University of British Columbia
6250 Applied Science Lane
Vancouver, BC
Canada V6T 1Z4

madalina@interchange.ubc.ca; desilva@mech.ubc.ca