Exploiting Structural Instabilities for the Development of Binary Logic and Neural Networks

Aaron John-Sabu* and Amuthan Arunkumar Ramabathiran†
Indian Institute of Technology Bombay, Mumbai, India

Structural instabilities such as buckling have presented challenges to the research community since the dawn of engineering due to their adverse effects on the integrity of assemblies and structures, from bridges and towers to airplanes and rockets. We present mechanical implementations of multistable elements that positively utilize buckling to physically emulate concepts from computer science and electrical engineering, such as digital circuitry and neural networks. We further provide systematic analysis into how these techniques can be expanded for broader applications in the domain of aerospace structures engineering.

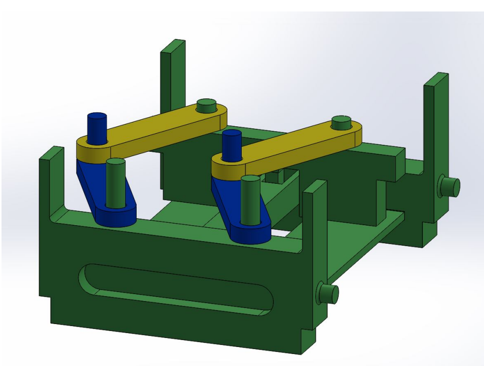
I. Introduction

Buckling is known to be detrimental to the integrity of structures, causing life-endangering situations on rail tracks, bridges, and other mechanical structures. This phenomenon typically causes failure when structures are loaded by compression. It has also been observed in [1] that there are several situations, especially in lightweight structures, in which instabilities, such as buckling, can be observed under tensile loads. As a counter to its adverse effects, we propose utilizing the phenomenon in structures for computational purposes. Firstly, we develop the equivalent of a logic element that is based on the concept of buckling. Particular combinations of this element with other similar and dissimilar elements lead to the development of blocks that represent the properties of logic gates, such as the NOT gate and the two-input AND gate. Secondly, we extend this application such that an equivalent of the neuron based on buckling is combined with similar and dissimilar units to form a neural network.

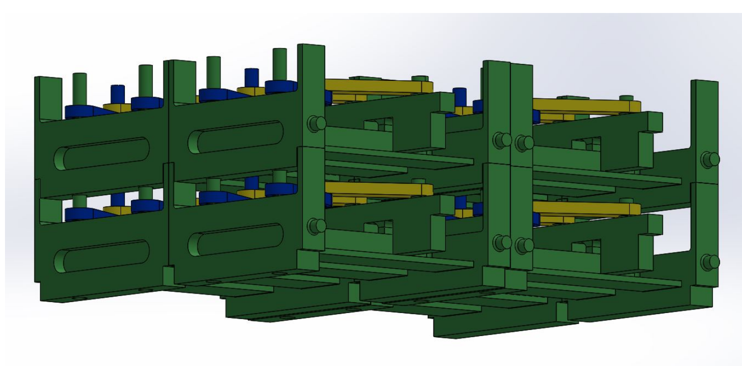
II. Logic Elements using Pseudo-Buckled Rods

A. The Unit Cell Mechanism

We consider a bistable element consisting of two interconnected rods attached on one end and movable along the lateral axis on the other end though constrained by an inbuilt spring. This is defined as a unit cell whose logic value is determined by its observed stable position (left = FALSE/0/LOW, right = TRUE/1/HIGH). This value is transmitted via



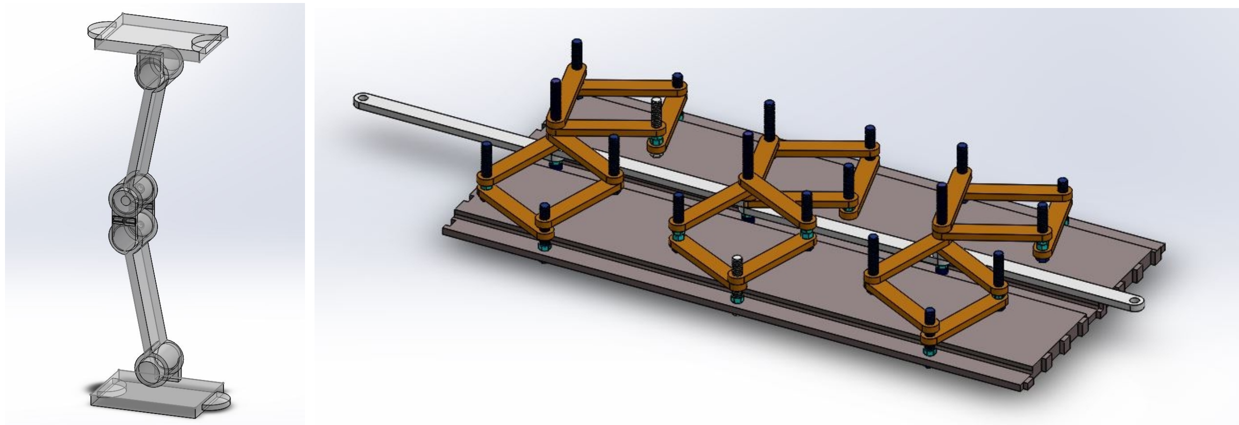
(a) Two-Cell Block



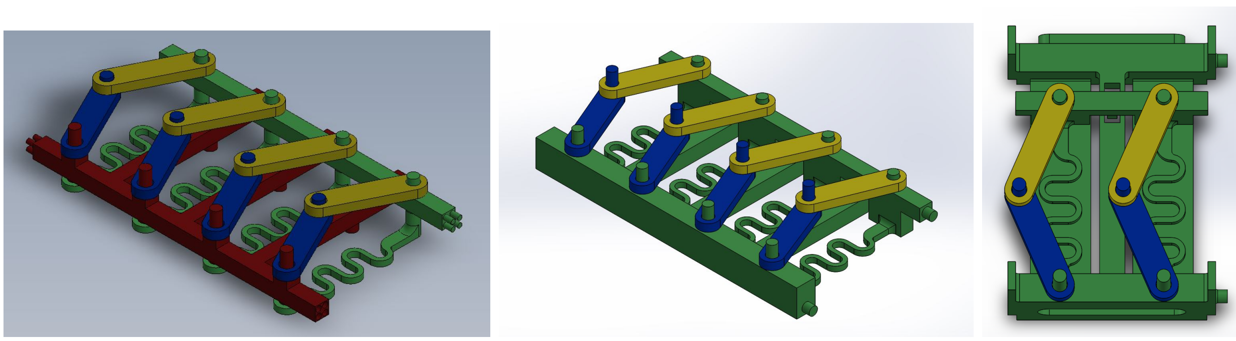
(b) Stack of 8 Blocks

Fig. 1 Design of the Unit Block

*Undergraduate Researcher, Department of Electrical Engineering, Indian Institute of Technology Bombay, Mumbai, India
†Assistant Professor, Department of Aerospace Engineering, Indian Institute of Technology Bombay, Mumbai, India



(a) Iteration 1 - 3D Unit Block **(b) Iteration 2 - 2D Unit Block**



(c) Iteration 3A - Hybrid Unit Block **(d) Iteration 3B - Hybrid Unit Block** **(e) Final Design**

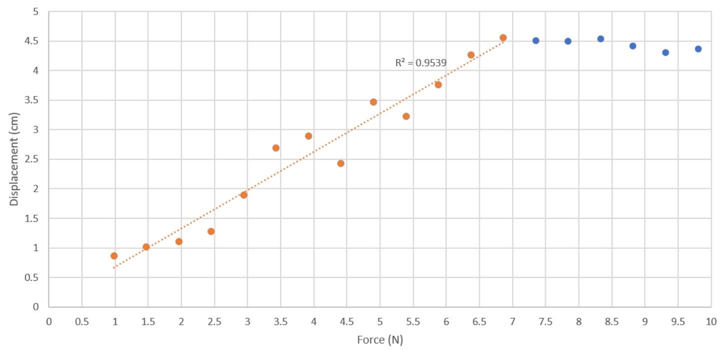
Fig. 2 Iterations of the Unit Cell

similar mechanical rods attached to the point of intersection of each bistable element, hence providing the possibility of creating a network out of several such unit cells with computational elements such as the previously proposed logic gates and neurons in between. The term ‘unit cell’ is to be understood as a single bistable element. However, for structural and aesthetic purposes, we place two or more unit cells in each unit block, a basic conglomerate of unit cells carrying a single value. Figure 2 represents various iterative designs for the unit block. Fig. 1a represents a single block consisting of two unit cells as per the final design, and Fig. 1b represents a stack of eight such blocks arranged as a cube.

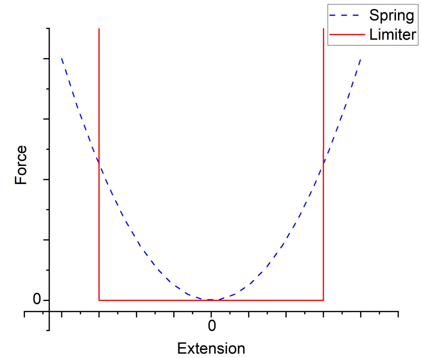
The developed unit cell can form a stable-unstable-stable spring-based rod mechanism. The structure of a cell is completed when it is attached to other such cells (for the transmission of information) or a gate structure (for the modification of information). The interlinking connectors can be springs or two-dimensional spring-shaped propagators for smooth transitions. However, on experimentation, we observe that rigid rods of suitable size support the same functionality, though motion will be stiffer in that case.

B. Compliance vs. Rigidity

Multistable elements have been developed in the past predominantly using compliant techniques as observed in [2–6]. In this research work, we generalize the concept of multistable elements by utilizing rigid unit cells that emulate the properties of a buckled rod and present techniques to develop logic elements from multiple such cells. This rigidity improves the reliability of the system and also makes the system more easily reproducible. Moreover, additive manufacturing of rigid materials is more feasible in terms of cost and availability. These features together make the mechanism both robust and scalable. The unit cell design incorporates an elastic part (‘unit-cell spring’) used to emulate the spring-like behavior of a buckled rod. This part is manufactured along with the unit cell from Poly-Lactic Acid (PLA) through the process of additive manufacturing and has been found to retain its elastic nature (Fig. 3b) despite multiple cycles of elongation and contraction.



(a) **Force - Displacement Rel. - PLA-based Spring**



(b) Force - Extension Rel. - Limiter vs. Spring

Fig. 3 Force Relations

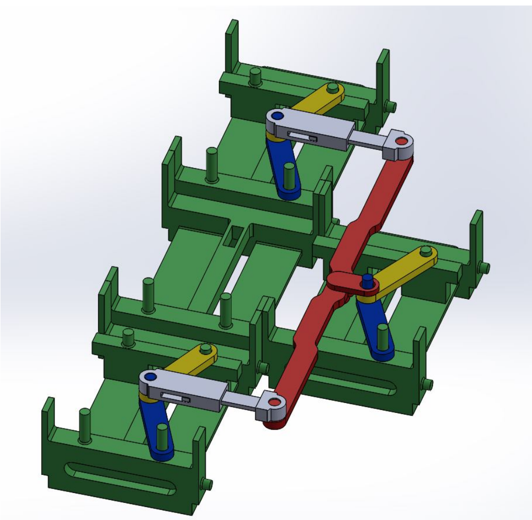
III. Logic Gates from Logic Elements

A. Functionally complete Logic Gates

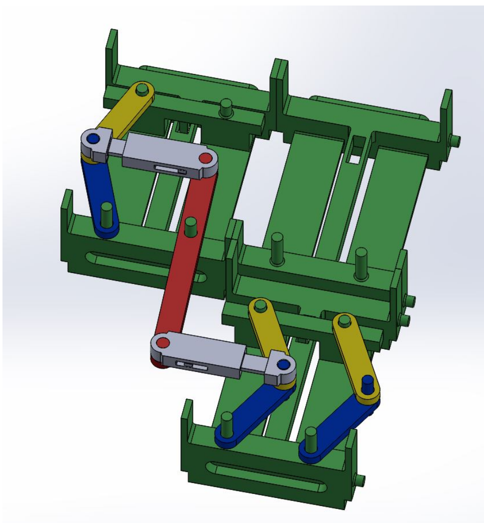
A functionally complete set of Boolean operators is a set of operators that can express all possible truth tables by combinations of the members of the given set. From our point of view, which predominantly deals with digital electronics, a functionally complete set of logic gates is a set of gates whose members can realize every possible logic gate. We consider functionally complete sets consisting of one element (the NAND gate or the NOR gate) and two elements (the AND and the NOT gates, or the OR and the NOT gates).

Although it is simpler to design a one-element functionally complete set of logic gates, the total number of logic elements is usually smaller when using two independent elements since a uniform representation of all the possible output combinations will demand fewer logic elements in the latter case. The design of more complex gates such as XOR and XNOR and larger circuits such as adders is also made much simpler when there are two independent gates. As a result, we have developed the AND and the NOT gates in the first half of this research.

B. Designing Logic Gates from Unit Cells



(a) AND Gate



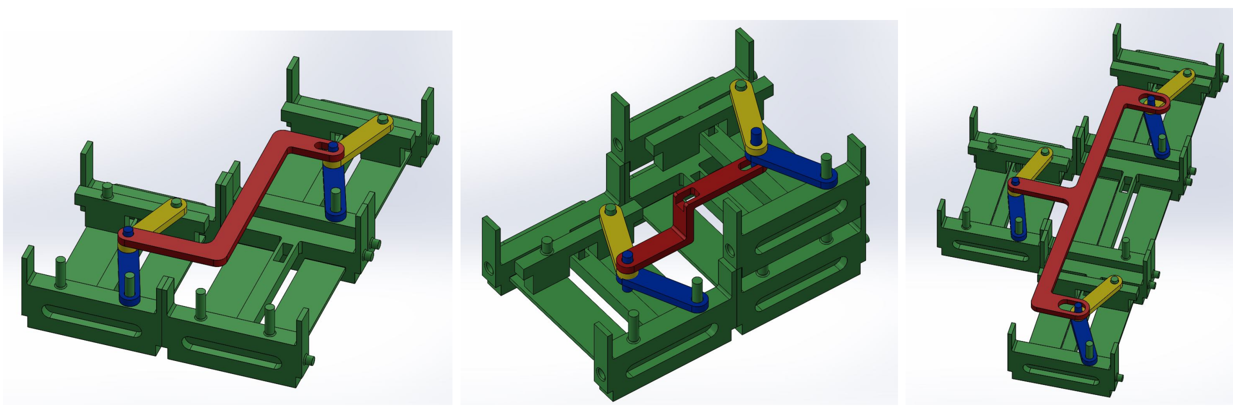
(b) NOT Gate

Fig. 4 Mechanical Design of Logic Gates

Due to the two-input nature of the AND gate, the operation is performed by transferring information from the input unit cells to the output unit cell. This is achieved with the help of two elements: the *limiter* and the *gate rod*. The limiter replaces a spring by providing some free space beyond which the two sides cannot be separated. The force graph of a limiter (in comparison to a spring) is represented in Fig. 3b. The gate rod performs the addition operation wherein the mechanical shift (representing the logic value) of each input cell partially translates the middle portion of the rod, thereby causing the entire rod to push the output bistable element to the HIGH state when both inputs are in the HIGH state. The design of the AND gate is depicted in Fig. 4a.

The NOT gate, unlike the AND gate, is a single-input gate and requires fewer components. The mechanical equivalent of the gate involves a lever mechanism to which the input and output logic elements are connected via limiters to prevent immobility. The design of the NOT gate is depicted in Fig. 4b.

C. Design-motivated Supplemental Blocks and Combinations

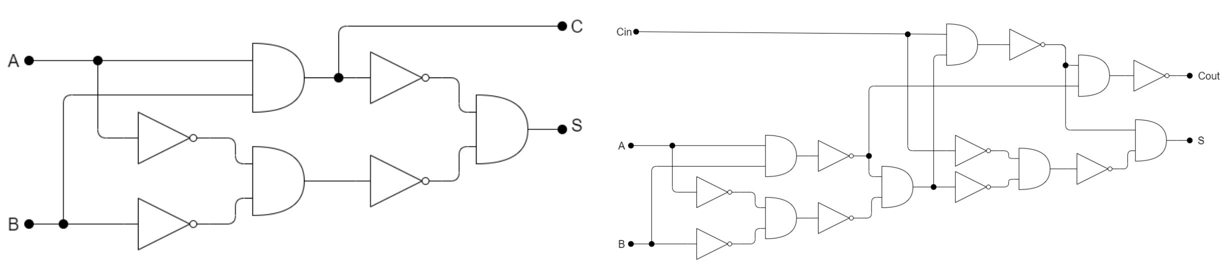


(a) +ve Y-axis Shifter **(b) +ve Z-axis Shifter** **(c) Splitter**

Fig. 5 Shifters

Blocks consisting of two unit cells can be used to translate information along the positive x-axis. However, they are not capable of transferring information along the y-axis or the z-axis. For this purpose, we propose the use of shifters (Fig. 5a, 5b) that transfer mechanical information in the required direction. Furthermore, we develop splitters as shown in Fig. 5c to split a single piece of data into two or more identical pieces. This helps facilitate the transfer of the same piece of data into multiple logic gates for complex circuits.

D. Expanding the Scope to Adders



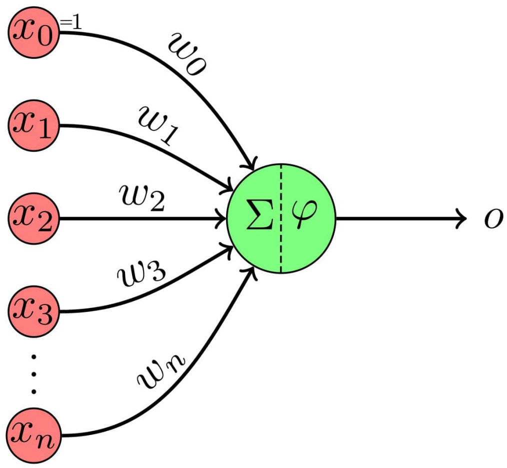
(a) Half-Adder **(b) Full-Adder**

Fig. 6 Adders from AND and NOT gates

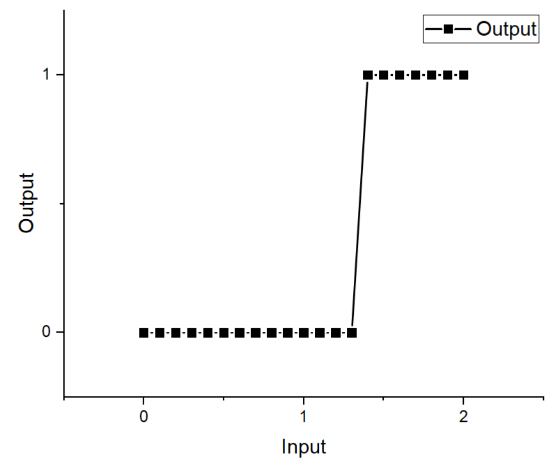
Logic gates can be arranged in a specific manner as shown in Fig. 6 to develop a multi-index adder that utilizes the binary form of the addends. Bringing in the basic concepts into play, for the addition of two integers, the ones' digit of any number (A_1) will be added with at most a single digit (B_1), hence the number of inputs being two, A_1 and B_1. Such an adder is known as a half-adder. On the contrary, any other digit in an integer (A_n) gets added with the corresponding digit of the second integer (B_n) and the carry-over (C_{n-1}) from the previous addition. Hence, such an adder will have three inputs - A_n, B_n, and C_{n-1}. This gives us a full-adder. Both adders will have two outputs: the sum (S_n) and the carry-over (C_n).

IV. Developing a Neuron for Binary Neural Networks

Following the development of mechanical logic gates, we develop a general-purpose form of programmable matter based on the mechanically bistable logic elements. We perform this by designing a mechanical neural network that can be trained offline and implemented online. Similar to the computerized version, the mechanical neural network consists of several neurons (or perceptrons) performing computations on multiple inputs and providing a single output. Here, the inputs and outputs are physically manifested as bistable elements, capable of flipping between two states representing 0 (false) and 1 (true).



(a) The Perceptron Unit



(b) Graph of the Activation Function

Fig. 7 Basis for Design of the Neuron

Computation on a neuron (Fig. 7a) can be divided into two parts - the linear superposition of the inputs, each multiplied by a weight (also implemented mechanically), and the application of an activation function on this intermediate value in order to make the output a non-linear dependence of the inputs. The activation function is modeled based on functions used in the theoretical neural network, such as the ReLU function and the sigmoid function. Such a non-linear function removes the redundancy that occurs when several layers are used in the network.

Linear superposition requires weights that may be incorporated by modifying spring constants. However, this is a challenging problem since it requires the input from a bistable element to be discretized into more than two values. Furthermore, we require a mechanical technique to replicate the process of addition. In order to solve these two challenges, we obtain inspiration from the 2D Unit Block (Fig. 2b) and develop unfixed four-bar mechanisms that have different bar lengths to incorporate the effect of weights and take in input from multiple inputs in order to add them. This method of linear superposition is represented by the multi-bar linkage mechanism in Fig. 9.

We utilize a non-linear activation as represented in Fig. 7b. In order to develop the mechanical equivalent of this function, we obtain the intermediate output from linear superposition in a special rod that connects to an output logic element. The placement of a screw (dark blue in the figure) determines the cutoff value for the activation function. This helps convert the non-binary intermediate output of linear superposition into a binary but non-linearly-dependent final output. We may develop the ReLU function for multi-input neurons by using a stopper mechanism in one direction while letting the input freely pass through in the other direction.

Occasionally, neural networks require that the neuron's output be replicated to use it for several neurons at the next layer. A mechanical replicator can be used to split the output from the neuron and distribute it several times. Here, we

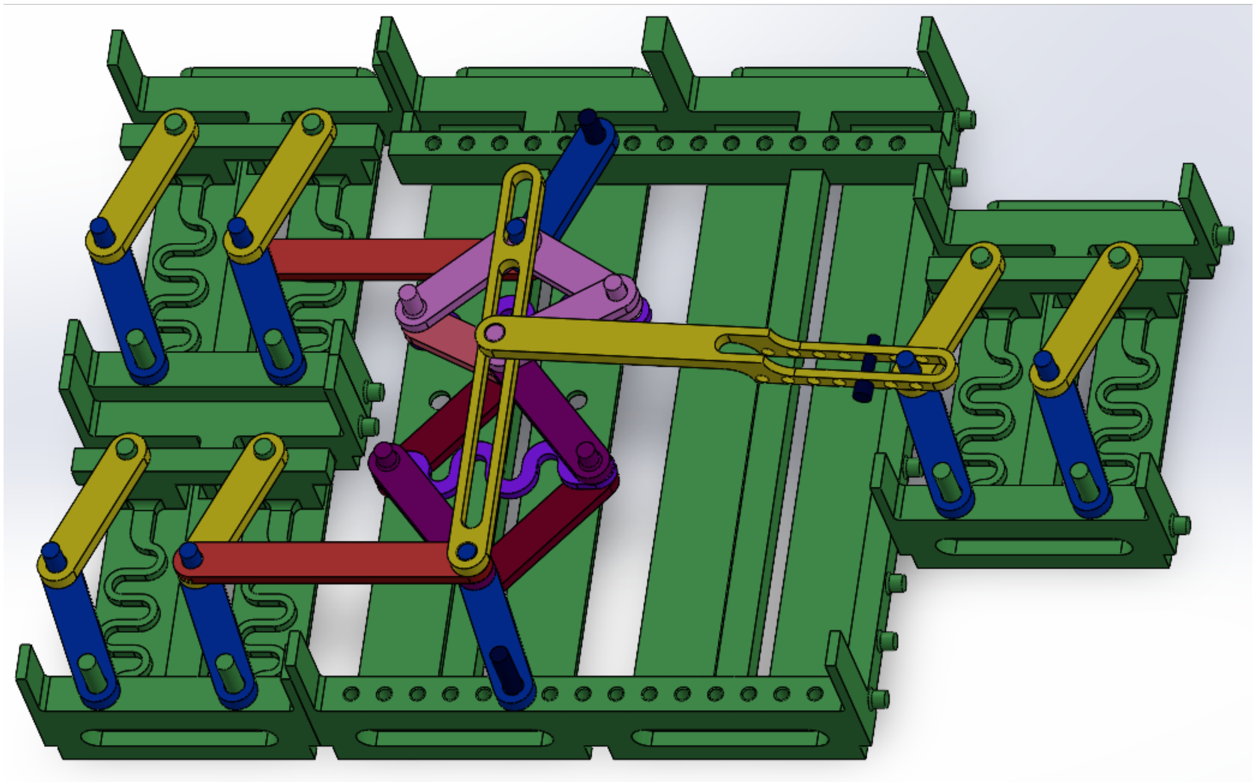


Fig. 8 The Mechanical Neuron

Fig. 9 Design of the Two-Input Neuron

incorporate the splitter design (Refer to Sec. III.C) depicted in Fig. 5c.

V. Conclusion

We have presented a general framework for the development of mechanical equivalents of logic gates and neural networks. The AND gate is designed using limiters that behave as infinite springs with hysteresis, and the NOT gate is designed using a lever mechanism. Furthermore, the sufficiency of these gates is proven by the concept of functionally complete logic gates. We develop neurons in the model of the perceptron unit. The two-input neuron includes a pair of unhinged four-bar linkages that add the two inputs and a specially designed rod for converting the non-binary intermediate output into a binary non-linear output.

While the direct application of this idea in real life is not a reasonable option, the scope of this field in concepts such as adaptive and smart materials is promising. For instance, a complex of logic gates could be arranged in a pre-determined fashion inside an elastic covering to behave as the wing of an airplane wherein inputs provided to the complex at the intersection of the wing and the fuselage determine the configuration and sweep of the wing.

Acknowledgments

The authors acknowledge the support of the Tinkerers' Laboratory (Institute Technical Council) and Prof. R. K. Pant and Mr. Raman Verma of the Lighter-Than-Air Systems Lab (Dept. of Aerospace Engineering), both at the Indian Institute of Technology Bombay, in the additive manufacturing of 3D models developed as part of this research.

References

[1] Rammerstorfer, F. G., "Buckling of elastic structures under tensile loads," *Acta Mechanica*, Vol. 229, No. 2, 2018, pp. 881–900. https://doi.org/https://doi.org/10.1007/s00707-017-2006-1.

[2] Chen, G., Gou, Y., and Yang, L., "Research on Multistable Compliant Mechanisms: The State of the Art," *Proceedings of the 9th International Conference on Frontiers of Design and Manufacturing (ICFDM 2010)*, 2010, pp. 17–19.

[3] Santer, M., and Pellegrino, S., "Compliant multistable structural elements," *International Journal of Solids and Structures*, Vol. 45, No. 24, 2008, pp. 6190–6204. https://doi.org/https://doi.org/10.1016/j.ijsolstr.2008.07.014.

[4] Wang, D.-A., Pham, H.-T., and Hsieh, Y.-H., "Dynamical switching of an electromagnetically driven compliant bistable mechanism," *Sensors and Actuators A: Physical*, Vol. 149, No. 1, 2009, pp. 143–151. https://doi.org/https://doi.org/10.1016/j.sna.2008.11.004.

[5] Opdahl, P. G., Jensen, B. D., and Howell, L. L., "An investigation into compliant bistable mechanisms," *ASME Design Engineering Technical Conferences*, 1998.

[6] Masters, N. D., and Howell, L. L., "A self-retracting fully compliant bistable micromechanism," *Journal of Microelectromechanical Systems*, Vol. 12, No. 3, 2003, pp. 273–280. https://doi.org/10.1109/JMEMS.2003.811751.