Delft University of Technology

Bi-threshold Gates for Mechanical Logic in Intelligent Metamaterials

Authors:
Eoinlee Bley (5216737)

Supervisors:
Dr. Davood Farhadi Machekposhti
Malte ten Wolde

in partial fulfillment of the requirements for the degree of

Master of Science
in Mechanical Engineering

September 20, 2023

Delft University of Technology

Abstract

Contents

0.1 Elementary Cellular Automata Formalism

1. State Space: $S = \{0, 1\}$
2. Neighborhood Configuration: N
   $N = (N_{-1}, N_0, N_1) \quad \text{where} \quad N_{-1}, N_0, N_1 \in S$
3. Rule Function: $f : S^3 \to S$
4. Rule Set: R
5. Cube Domain: $D \subset \mathbb{R}^3$
   Each vertex directly corresponds to a neighborhood configuration N, and its state is
6. Separating Planes: P
   Defined by a single normal vector $\mathbf{n}$ and different offsets $\{d_1, d_2, \ldots, d_n\}$.
7. Domain Classification Function: $\Delta : D \to \{0, 1, 2, 3\}$
   $\Delta(x) = \sum_{i=1}^{n} H(n_x \cdot x_x + n_y \cdot x_y + n_z \cdot x_z - d_i)$
   $H(z) = \begin{cases} 0 & \text{if } z < 0 \\ 1 & \text{if } z \geq 0 \end{cases}$

0.2 Wolfram Numbering Scheme for ECA

In the Wolfram numbering scheme for Elementary Cellular Automata (ECA), the rule set R can be uniquely identified by a single integer, which is the binary representation of the output states for all possible neighborhood configurations. For Rule 110, the binary representation is formed by considering all 8 possible 3-cell neighborhood configurations, starting from 111 down to 000.
For example, in Rule 110, the corresponding output states for these configurations are 01101110. Here's how it maps:

Neighborhood Configuration	Output State	Binary Position (b)
111	0	b_7
110	1	b_6
101	1	b_5
100	0	b_4
011	1	b_3
010	1	b_2
001	1	b_1
000	0	b_0

So, the Wolfram number for Rule 110 is obtained by reading the output states from b_7 to b_0 as a binary number: $01101110_2 = 110_{10}$.

1 Introduction

2 Results & Discussion

Geometric Representation of Cellular Automata Rules

Cellular automata (CA) are grid-based computational models where each cell evolves over time according to a rule set R. In Elementary Cellular Automata (ECA), the domain is one-dimensional and the state space is binary, $S = \{0, 1\}$. Each cell's future state is determined by its current state and those of its immediate neighbors.
Mathematically, for cell i at time t, the next state u_i^{t+1} is governed by a rule function $f : S^3 \to S$:

$$u_i^{t+1} = f(u_{i-1}^t, u_i^t, u_{i+1}^t)$$

With a binary state and 3-cell neighborhood, there are $2^8 = 256$ unique ECA rules. These are indexed from 0 to 255, following Wolfram's convention.
For example, Rule 110 is defined as:

$$f_{110} : (0, 0, 0) \to 0,\ (0, 0, 1) \to 1,\ \ldots,\ (1, 1, 1) \to 0$$

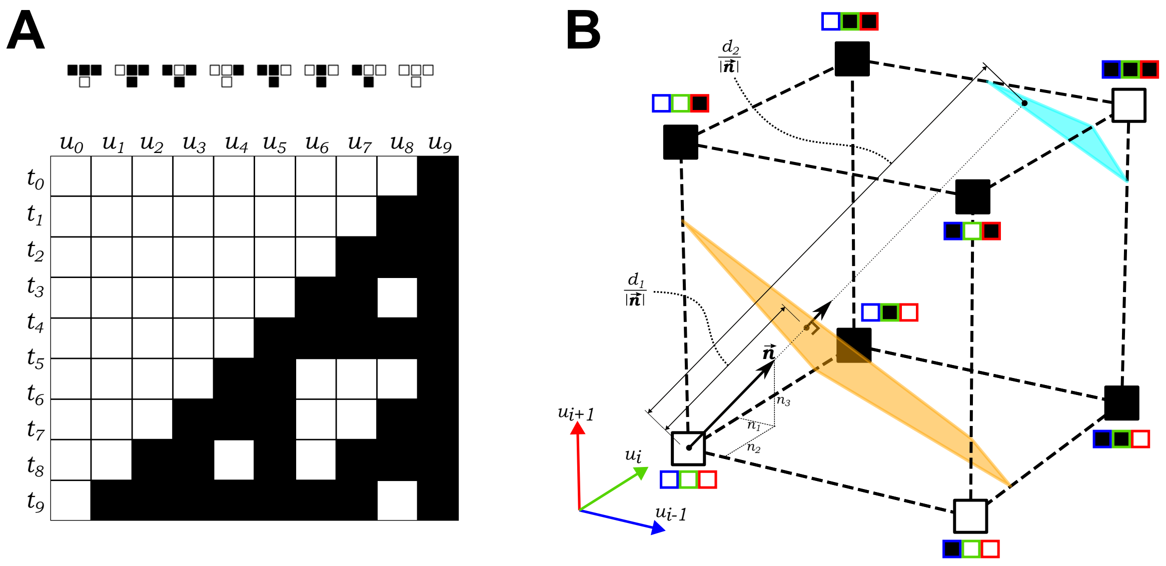

Figure 1: A. The transition rule and time evolution of the Rule 110 cellular automata. B. Cube representation Rule 110 with separating planes defined by normal vector $\overrightarrow{n}$ and offset constants d_1 and d_2.

Concept Mechanism

Working Principle

Simulation

3 Conclusion

4 References

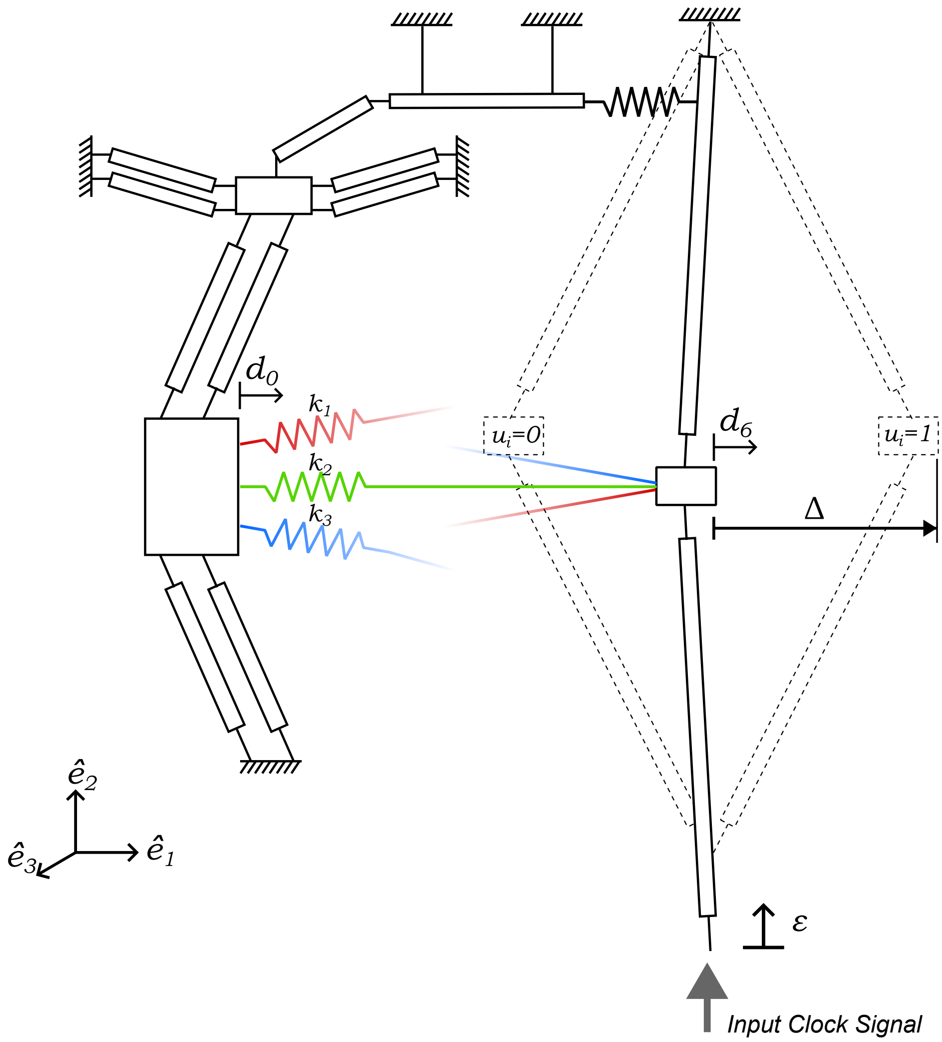

Figure 2: This is a figure.

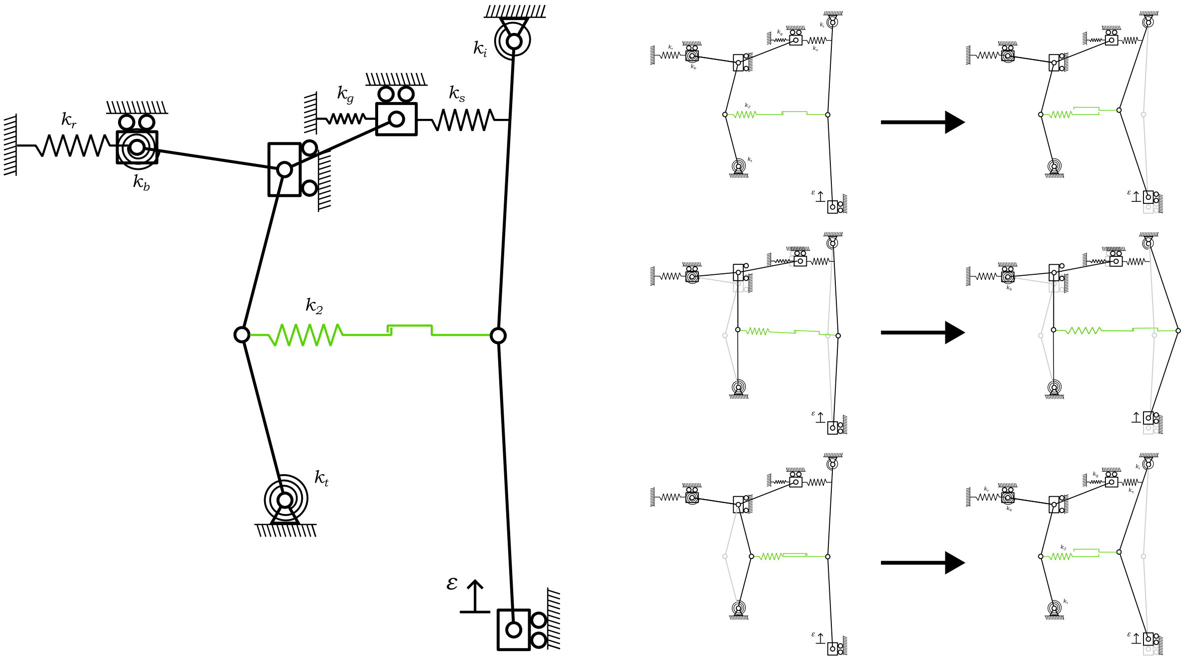

Figure 3: This is a figure.

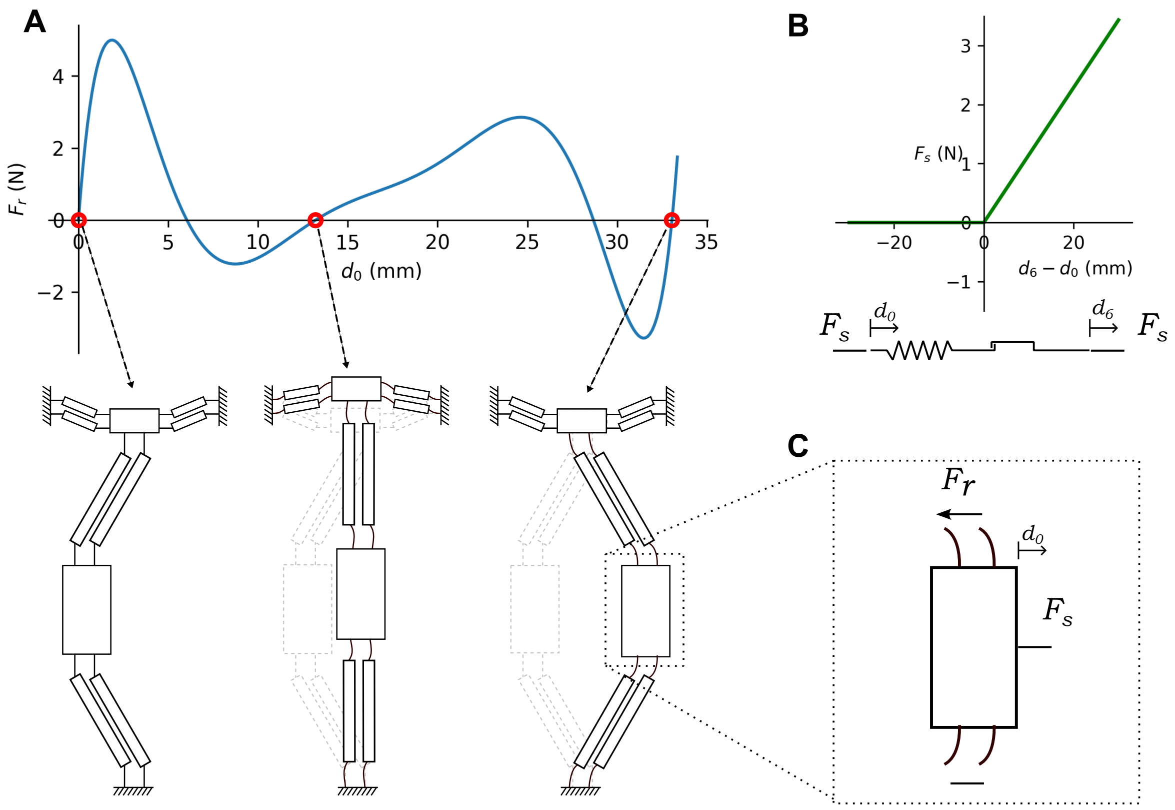

Figure 4: This is a figure.

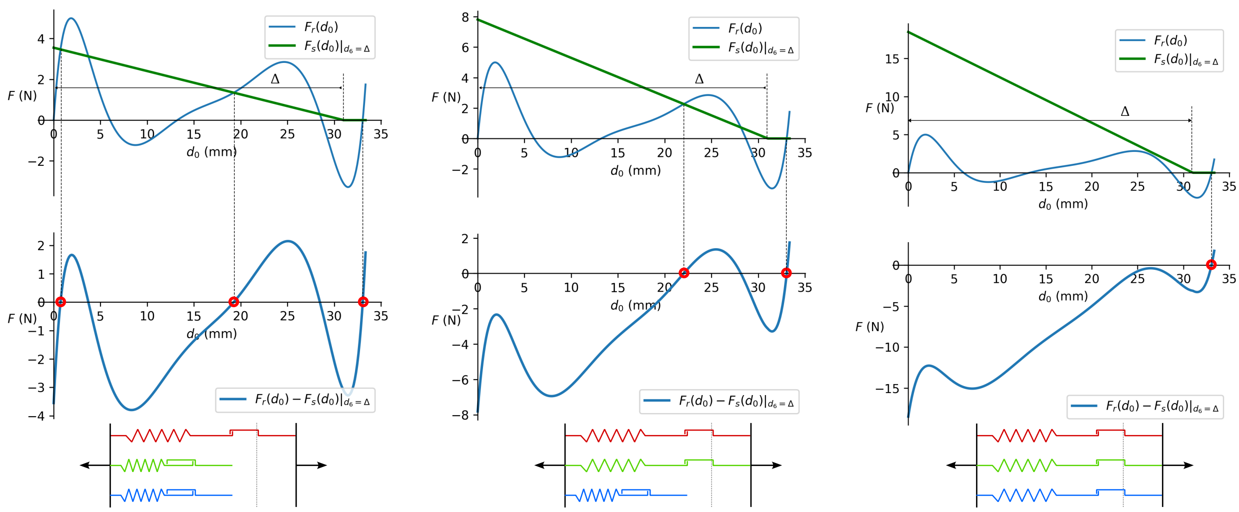

Figure 5: This is a figure.

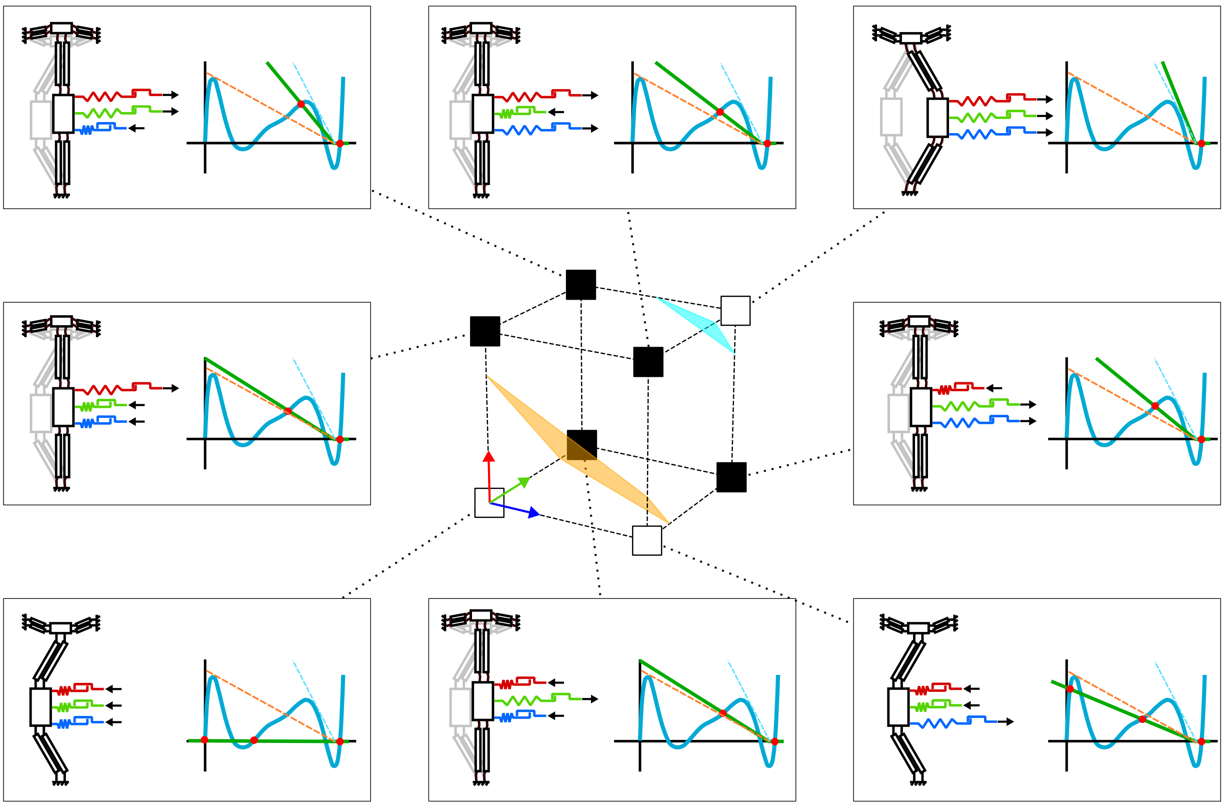

Figure 6: This is a figure.

A Supplementary Material

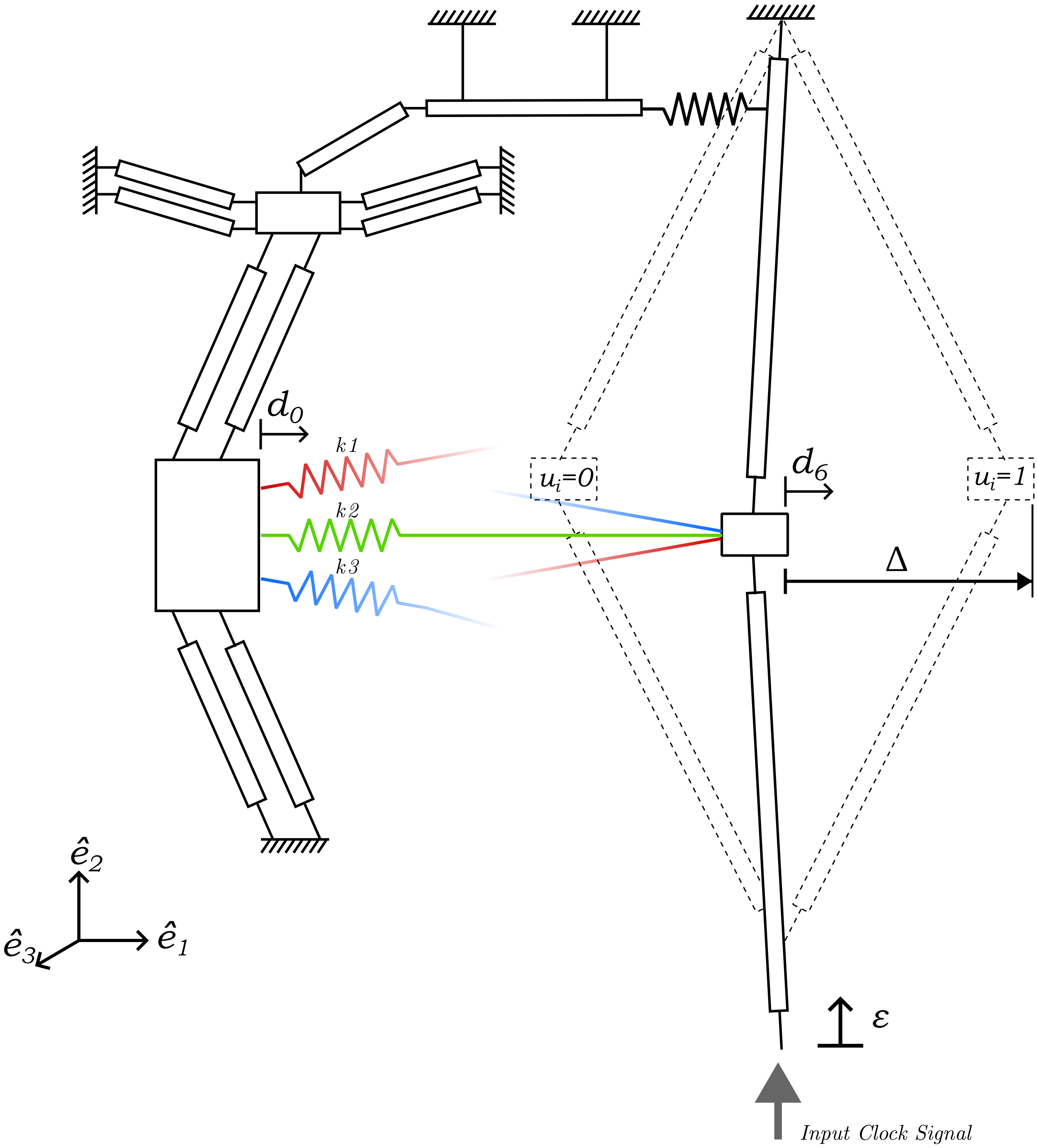

Figure 7: 1D Cellular Automata