

# A Toolbox for Designing 3D-Printing-Ready Pneumatic Circuits for Controlling Soft Robots

*Sihan Wang,\* Liang He, Yao Yao, Chenying Liu, and Perla Maiolino*

The robotics field has witnessed a rapid increase in the popularity of soft pneumatic robots due to their adaptability to unstructured environments and safe human–robot interaction. However, many of them are still controlled by rigid solenoid valves and microcontrollers. Recent research works have introduced various electronics-free computational devices for soft robotic applications. Yet, connecting these individual components into pneumatic circuits still remains challenging. The manual circuit design, fabrication, and piping process can bring inefficient circuit layout, redundant tubing, unreliable fittings, and potentially human mistakes. This work presents an open-source toolbox for automatically designing pneumatic circuits for soft robotic applications. The toolbox takes the desired computation function along with some space constraints as inputs, and generates a 3D-printable file of a pneumatic circuit that can be fabricated and used directly. This is achieved by efficiently positioning the logic gates and air channels while considering the workspace constraints, fabrication limits, material cost, and system complexity. The versatility of the design automation algorithm has been tested with two different truth tables and six different workspaces. The work also showcases utilizing the toolbox to design a computational circuit that governs the movements of a soft robotic hand with four pneumatic actuators.

advantages in cost, manufacturability, and safety for human–robot interaction. Previous research has shown that soft pneumatic robots can execute various challenging tasks with much simpler control strategies compared to their rigid counterparts, such as locomotion over different terrains,<sup>[2,3]</sup> multimodal object grasping,<sup>[4,5]</sup> navigating through complex environments,<sup>[6]</sup> and so on. Despite the simplified control strategy and low computational cost required, many soft robots are still controlled by conventional rigid electronic systems such as rigid valves, pumps, microcontrollers, and motor drivers.<sup>[7,8]</sup> Although these devices provide high energy density, computation power, and fast execution, there are certain problems when applying them to soft robots. First, using solenoid valves means that both electrical and pneumatic sources need to be placed onboard the robot. These rigid components significantly sacrifice the benefits of being soft. The unnecessary conversion between pneumatic signals and electric signals also brings additional system complex-

## 1. Introduction


There has been a surge of research interest in soft pneumatic robots.<sup>[1]</sup> By exploiting the interaction between pressurized air and soft elastomeric materials, the robots come with high compliance and adaptability. This mechanism also brings other

ity. Second, from the manufacturing point of view, the electronic system comes with completely different fabrication techniques used by soft pneumatic actuators (e.g., silicone casting, 3D printing). This means that it's not easy to integrate these devices into the body of soft robots. Moreover, there are certain extreme scenarios where the electrical circuits cannot be used, such as the nuclear plant exploration with high radiation,<sup>[9]</sup> tissue interaction during metal-free scenarios in magnetic resonance imaging,<sup>[10]</sup> and so on. Therefore, developing pneumatic circuits that can be used to control soft robots holds significant importance. It enables us to integrate the computation and control systems into the body of soft robots while keeping the adaptability and compliance brought by soft materials.

State-of-the-art works have shown various approaches to integrating pneumatic sources into soft robots, such as chemical decomposition,<sup>[11,12]</sup> CO<sub>2</sub> cartridges,<sup>[13]</sup> and latex balloons.<sup>[14]</sup> Meanwhile, different soft pneumatic devices have been developed to replace the traditional microcontrollers for signal computation.<sup>[15–17]</sup> Entirely soft silicone cast valves with a semi-spherical bistable membrane and kinking tubes is able to achieve multiple computation functions including digital logic,<sup>[18,19]</sup> ring oscillator,<sup>[20]</sup> and nonvolatile memory.<sup>[21]</sup> The silicone-cast nature brings these devices high durability, low cost, and most importantly high compliance so that the devices can be integrated into

S. Wang, L. He, Y. Yao, C. Liu, P. Maiolino  
 Department of Engineering Science  
 University of Oxford  
 Oxford OX1 3PJ, UK  
 E-mail: sihan.wang@eng.ox.ac.uk

P. Maiolino  
 DIMEC Department of Mechanics and Machine Design  
 University of Genoa  
 16145 Genova, Italy

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/aisy.202300394>.

© 2023 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.1002/aisy.202300394

various soft robots.<sup>[13,22]</sup> Various different designs based on soft kinking valves have been investigated to reduce the labor cost involved during the fabrication. Tube-balloon logic valves comprising straws, balloons, and PVC tubes have been used to perform different digital logic computations.<sup>[23]</sup> These devices can be easily fabricated with everyday objects and hold pressure up to 200 kPa. The kinking-tube mechanism has also been applied to develop logic-enabled textiles for smart robotic clothing.<sup>[24]</sup> More recently, researchers have designed a new type of electronics-free device based on a piston actuator and a bistable pneumatic switch.<sup>[25]</sup> These devices are capable of digital logic, controlled oscillation, memory storage, and analogue pressure regulation. These devices require less than 12 min of handcrafting for fabrication and can hold pressure up to 165 kPa.

All devices listed above require a significant level of design expertise and manual work during their applications. Though the labor cost for fabricating each component decreased significantly in more recent studies,<sup>[23,25]</sup> combining these components into computational circuits for real soft robotic applications still remains challenging in many aspects. For instance, the circuit schematics must be manually designed to achieve the desired computational function. Bringing the computational circuit onboard within the limited space in the robot body also requires users' knowledge and experience. Apart from the manual circuit design process, a manual piping process is required to securely and efficiently connect all individual units together. Each individual computation unit in the circuit is often highly similar in appearance and only distinguishable from each other based on some detailed configurations. This practical difficulty brings a high chance of human error. Problems related to redundant tubing and unreliable fittings can also affect the efficiency, response time, and durability of the systems. The abovementioned limitations are increasingly problematic as the computation circuit becomes more complex. This raises the need for an automated toolbox that directly transforms the user-specified robot behaviors into a ready-to-use computational pneumatic circuit that fits within a certain set of space constraints. By replacing the manual circuit design, fabrication, and piping process, the toolbox can significantly enhance the accessibility of electronics-free computational pneumatic circuits. Similar design automation tools which convert the users' demands into a ready-to-manufacture file have already been developed in other fields. Numerous electronics design automation tools based on VHDL & Verilog have been developed in the electronics industry.<sup>[26]</sup> Electromicrofluidic circuitry and genetic circuitry also see such design automation software.<sup>[27,28]</sup> However, such a toolbox is yet to be done for pneumatic circuits in soft robotics.

There are several existing software aiding the design of pneumatic circuits. Both PneuDRAW (SMC, Japan) and FluidDRAW (FESTO, Germany) can be used for drawing pneumatic circuits concisely and easily.<sup>[29,30]</sup> Moreover, FluidSIM (FESTO, Germany) enables users to simulate the performance and functions of a pneumatic circuit given its schematics.<sup>[31]</sup> However, although the software listed above simplifies the manual design process, none of them can design a ready-to-use pneumatic circuit based on the desired input-output mappings. In other words, achieving full design automation still remains challenging. More recent work on a web-based pneumatic circuit design tool makes a step toward this goal.<sup>[32]</sup> The

web-based toolbox takes the user's desired robot behavior in the form of a HIGH/LOW truth table and outputs a schematic guiding the wiring configuration. The validity of the soft compiler has been demonstrated using a pneumatic glove to implement several soft circuits composed of soft, bistable valves. This toolbox successfully automates the circuit design process, while the manual fabrication and piping processes still exist. The migration from the software-based pneumatic circuit to its physical implementation remains unsolved due to the limitations brought by the fabrication method of those silicone-cast devices. Moreover, the work is only capable of dealing with single-output truth tables.

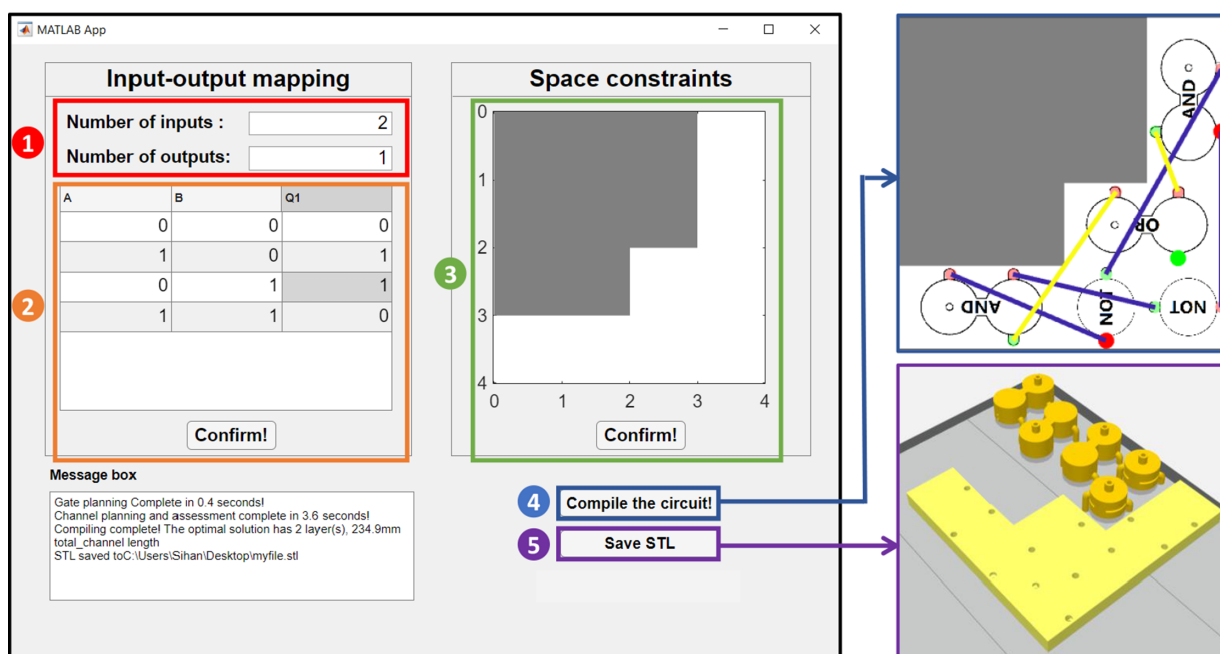
Multimaterial 3D printing technology can be a solution to convert conceptual circuit schematics into efficient physical prototypes without requiring manual fabrication and circuit piping. The technology enables users to print materials with different hardness seamlessly and with high accuracy. Previously, researchers designed a set of 3D-printable pneumatic circuit components (e.g., diode, normally "ON" transistor, normally "OFF" transistor) based on polyjet multimaterial 3D printing.<sup>[33]</sup> These devices can generate not only periodic signals for controlling a soft robotic turtle but also preprogrammed, aperiodic signals to control a robotic hand with multiple fingers. 3D-printable valves with bistable structures have also been exploited in previous works.<sup>[34,35]</sup> The preciseness and repeatability brought by 3D printing enable users to fine tune the snap-through pressure of the valves by adjusting several key design parameters. It also enables multiple independent air channels to be embedded in a single valve to reduce the number of units in a circuit.

To fill the gap, in this article, we present an automated design toolbox to generate 3D-printable pneumatic computational circuits. The toolbox comes with a graphical user interface (GUI), shown in **Figure 1**, to take users' desired computation functions in the form of a truth table and a set of 2D space constraints. It then schedules the pneumatic circuit which performs the desired computation within the space constraints and produces a 3D-printable CAD file that can be fabricated and used directly. This capability eliminates the manual work that was previously required during circuit design, fabrication, and piping processes, therefore enhancing the accessibility of pneumatic circuits for soft robotic applications. For validation, two different truth tables and six different sets of space constraints were input into the toolbox. The pneumatic circuits scheduled by the toolbox were then printed and tested in terms of their computation correctness, response time, and durability. An example of using the toolbox to design a circuit to control a four-finger soft robotic hand is also presented.

## 2. Methodology

### 2.1. Modular Design of the Logic Gates

To build the proposed pneumatic circuits for controlling soft robots, the fundamental hardware components would be a collection of pneumatically driven logic gates which have sufficient pressure tolerance, flow rate capability, and durability. At a minimum, one would require the design for NOT, AND, and OR gates to accomplish any user-specified truth table. This section



**Figure 1.** The user interface of the toolbox in MATLAB. The operation procedure is ordered here for clarity. ① indicates the region where users define the number of inputs and outputs of their desired computation. ② refers to the editable table where users define the truth table. ③ refers to the editable map where users specify the 2D space constraints of the pneumatic circuits. ④ and ⑤ refer to the buttons for planning the pneumatic circuit and saving the 3D-printable CAD files, respectively.

presents the hardware design for the logic gates employed in this work.

### 2.1.1. Design

The pneumatic logic gates used in this project are built upon an existing design of 3D-printed transistors,<sup>[33]</sup> which include a normally “ON” transistor and a normally “OFF” one. We reduce the size of both transistors for higher computation density and simplify the structure of the “OFF” transistor by removing an extraneous membrane. The pneumatic “NOT” gate is constructed using the modified normally “ON” transistor, the pneumatic “AND” gate is constructed by combining two normally “OFF” transistors in series, while the pneumatic “OR” gate is constructed by combining two normally “OFF” transistors in parallel. Moreover, the input and output ports of all logic gates are diverted to the bottom plane, so that all gates could be connected together into a circuit with a compact layer channel. The detailed designs are presented in **Figure 2A–C**.

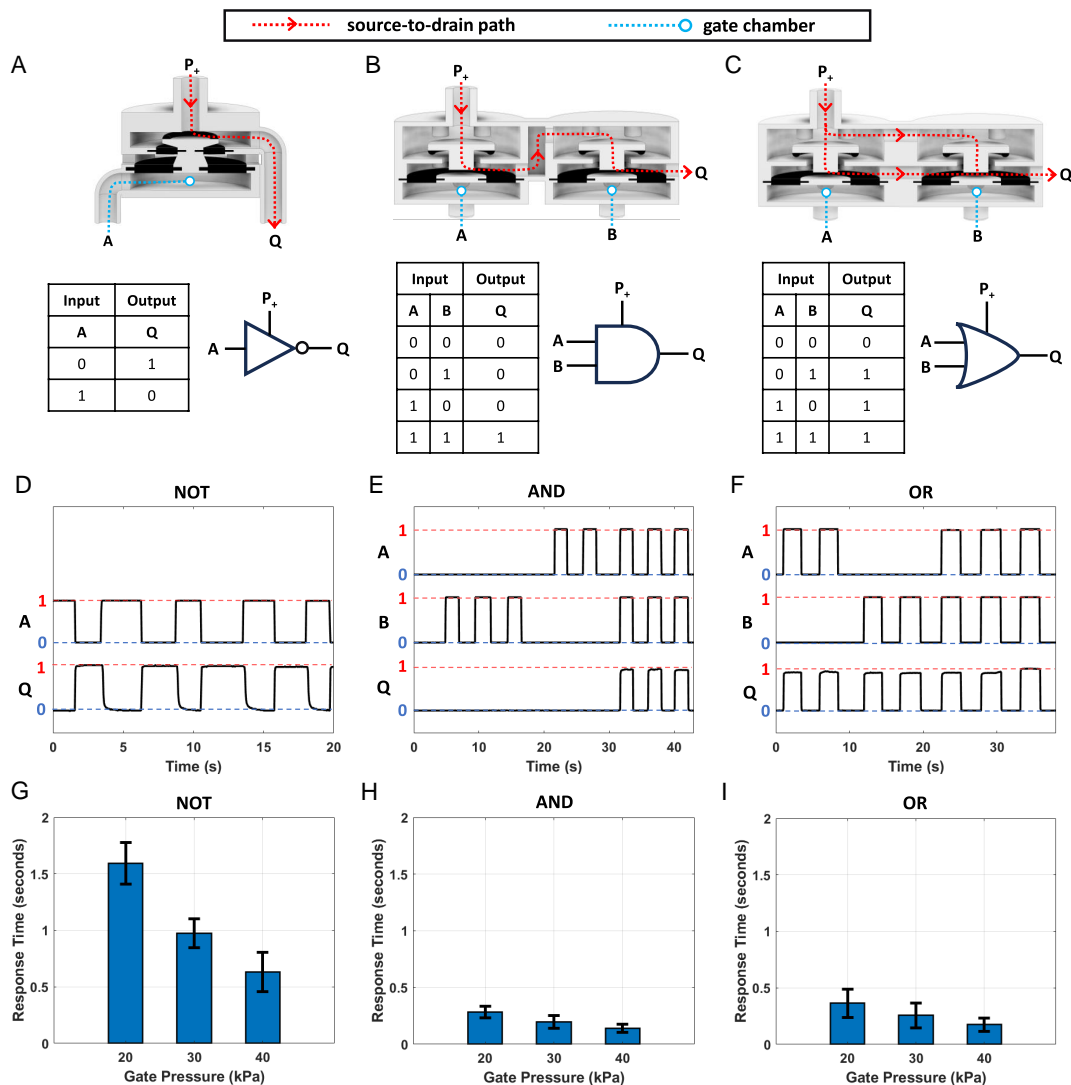
### 2.1.2. Characterization

To validate the fact that the logic gate design can be used by the proposed toolbox for pneumatic computational circuits, the computation correctness and the response time of these devices were experimentally characterized. The computation correctness refers to whether the device complements the desired input–output mappings, while the response time is the time difference from the engagement of input signals to the full development of the output signal. The obtained results are presented in

Figure 2D–I. All three logic gates perform the computation as desired. Response time, measured as the average time duration between the engagement of pneumatic inputs and the full development of pneumatic outputs, is used to describe the responsiveness of these logic gates. It was observed that the logic gates bring a faster response when the pneumatic signal amplitude increases since the larger driving force can deform the soft membrane in the logic gate faster despite its viscoelasticity. The response times of the NOT, AND, and OR gates with 40 kPa pneumatic signals are 0.63, 0.15, and 0.18 s respectively. This inconsistency in response time is caused by the different mechanisms used in different gates. The operation of the NOT gate requires a relatively large deformation from its soft membranes so that the compliant O-ring can block or release the air channel. The viscoelasticity of the soft materials therefore significantly slows down the response of the NOT gate. While in AND and OR gates, the regulation of the air channel is achieved by the movements of a free-floating sealing disc, which is much faster. The required deformation of the soft membranes is also smaller, contributing to the shorter response time observed in AND and OR gates.

## 2.2. Design Automation of the Pneumatic Computational Circuit

The objective of the proposed toolbox is to automatically convert the desired computation functions into a 3D-printing-ready CAD file that fits within a 2D workspace defined by the user. The computation functions are given in the format of a multi-input, multi-output truth table, while each input and output is a digital pneumatic signal. This truth table format can represent the

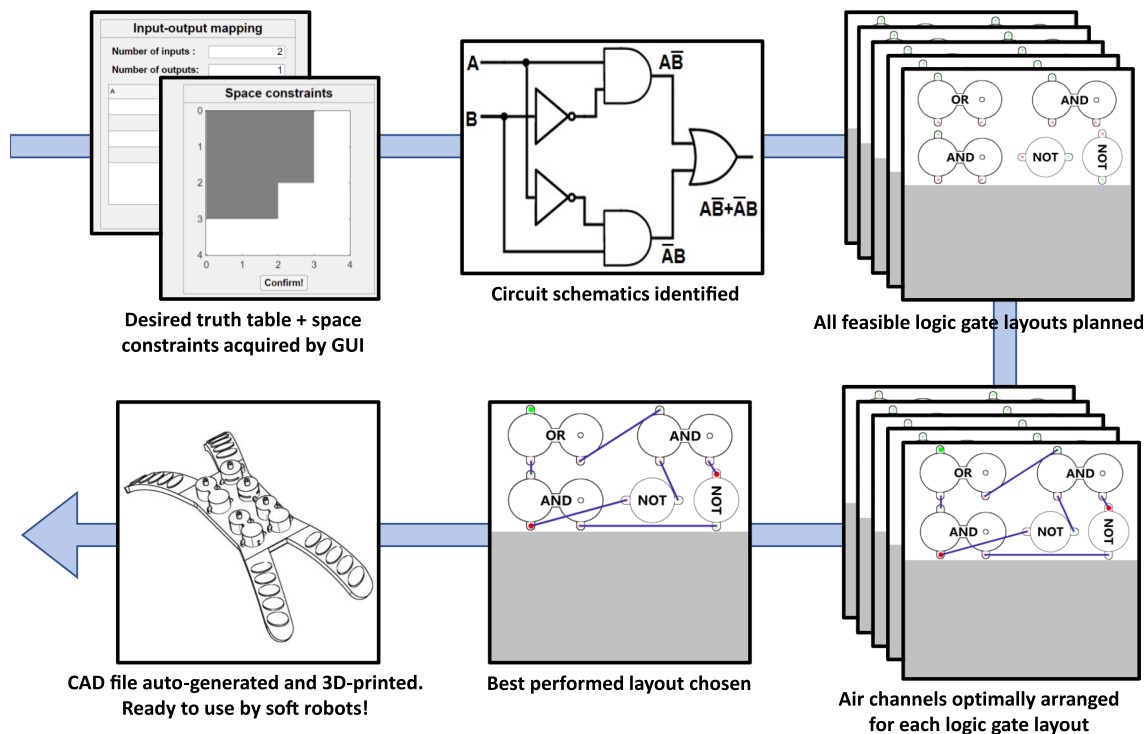


**Figure 2.** A) The structure and the working principle of a pneumatic NOT gate. B) The structure and the working principle of a pneumatic AND gate. C) The structure and the working principle of a pneumatic OR gate. The design of these gates has been influenced by and adapted from a prior work.<sup>[33]</sup> D–F) The pressure–time curve of the three logic gates at their inputs and outputs. G–I) The response time of each logic gate under pneumatic inputs with different amplitudes.

realization of various meaningful robot behaviors. The output of the compiler is in the format of a 3D-printing-ready CAD model. The CAD model can be printed out using multimaterial 3D printers with soluble supports. This enables rapid, accurate, and repeatable manufacture of pneumatic circuits for controlling a wide range of soft robots, with a minimum amount of manual work involved. Moreover, the 3D-printable pneumatic circuit can be printed inside the soft robot body, opening up the potential for an untethered robotic system with onboard computation. The proposed toolbox is based on MATLAB and Comsol. The former is used for building GUI and compiling circuit payout, while the latter is used for converting the planned layouts into 3D-printable CAD files. The user interface of the toolbox is illustrated in Figure 1, while Figure 3 provides an overview of the circuit planning procedure. The subsequent sections will delve into the details of this procedure.

### 2.2.1. Circuit Schematics Identification

The first step during compilation is to convert the user-input truth table into a schematic that tells the minimal type/number of the logic gates required and how the gates should be connected into a circuit to achieve the desired computation. The input truth table is initially converted into logic expressions via the sum-of-product algorithm. For each row in the truth table with a HIGH output, the SOP algorithm generates a logic expression based on the inputs in that row. It then combines all these logic expressions with OR operators. The raw output of the SOP algorithm is often highly complex. The inefficient logic expressions bring redundant logic gates and air channels into the circuit increase the circuit size/weight, slow down the circuit response, and bring additional complexity into the following compiling process. To address these issues, the logic expressions are simplified to



**Figure 3.** An overview of the pneumatic computation circuit design toolbox that converts the desired computation functions into a 3D-printing-ready CAD model directly. The toolbox enables rapid design and manufacture of the computational pneumatic circuits for a wide range of soft robots, with minimum manual work involved.

their simplest forms via MATLAB Symbolic Toolbox to minimize the number of logic gates used. The simplified expressions are then converted to postfix notation for the ease of further processing.

A customized data structure “PneuSignal” is used to represent the circuit schematic. Each “PneuSignal” describes the information about an air channel in the circuit (may contain forks) located between logic gates or the circuit inputs/outputs. All properties inside a “PneuSignal” and their meanings are described in **Table 1**. By identifying all “PneuSignals” involved in a pneumatic circuit, the whole schematic is effectively obtained. The algorithm used to convert the postfix logic expressions into all “PneuSignal” involved in the circuit is described in Supporting Information S2.

### 2.2.2. Logic Gate Layout Planning

The logic gate layout determines the placement (e.g., position, orientation) of each logic gate within the 2D workspace. The

**Table 1.** Detailed properties inside the “PneuSignal” data structure. Each “PneuSignal” represents them.

| Properties | Meaning                                                       |
|------------|---------------------------------------------------------------|
| Value      | Logic expressions of this pneumatic signal                    |
| Up         | The logic gate located upstream of this pneumatic signal      |
| Down       | The logic gate(s) located downstream of this pneumatic signal |

backtracking algorithm is used to find out all possible logic gate layouts that fit within the user-input 2D workspace. This is essential for ensuring that the best solution is ultimately found after the planning and assessment of air channels. The usage of the backtracking algorithm is highly common in compiler design due to its simplicity in programming, low space complexity, and compatibility to solve almost all kinds of problems.<sup>[36,37]</sup> To simplify the problem and reduce the algorithm complexity, the user-input 2D space constraints are modeled as an  $m \times n$  matrix within which only certain entries are available for gate placement. Each AND/OR gate requires two adjacent available entries to be placed while each NOT gate requires only one available entry. The detailed implementation of the backtracking method for logic gate layout planning is described in Supporting Information S3.

### 2.2.3. Air Channel Layout Planning

The objective of this step is to plan the air channel layout for each valid logic gate layout obtained in the previous steps. The air channels are designed to connect all logic gates and inputs/outputs in the circuit together according to the circuit schematic. The Bentley–Ottmann algorithm is used here to detect any intersected air channels with the lowest complexity.<sup>[38]</sup> These intersected channels will be placed at different layers to ensure each pneumatic signal stays independent from the others. The greedy algorithm is then used to generate the optimal air channel arrangement which has the least total number of layers while

ensuring no intersected channels are within the same layer. Note that each different logic gate layout has its own optimal channel layout, which will be evaluated in the following step.

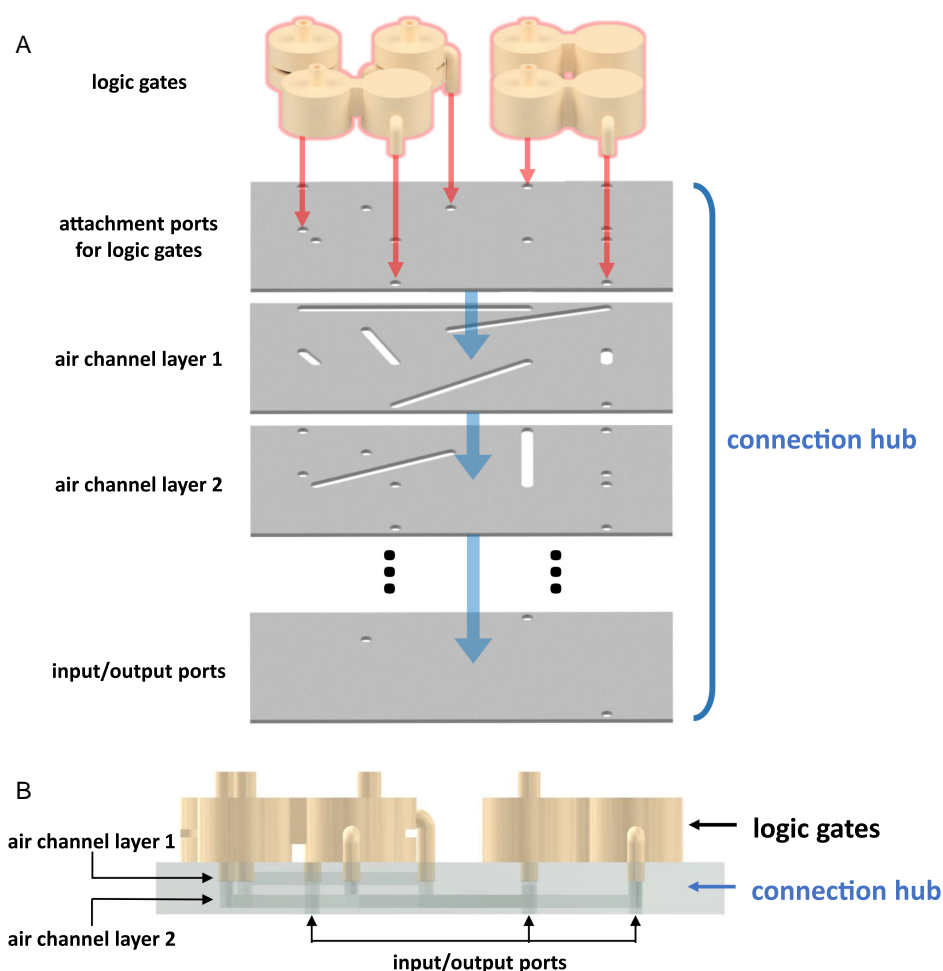
#### 2.2.4. Optimal Layout Assessment

The candidate solutions obtained from previous steps are examined through the following aspects: 1) The minimum distance between any pair of independent channels must be larger than a threshold value to prevent signal interference caused by leaking. This threshold value depends on the manufacturing precision and required robustness of the pneumatic circuit. This acts like a solid constraint, meaning that any pneumatic circuit which fails to satisfy this criterion will be rejected for further assessment; 2) The optimal solution should have the least number of layers. Any unnecessary addition in the total layer number thickens the overall system, increases the fabrication time/cost, and slows down the system response; and 3) The optimal solution should have the lowest total air channel lengths. This reduces the internal volume within the computational systems, therefore minimizing the response time and energy

consumption during operation. It also reduces the time required for postprinting support dissolving.

#### 2.2.5. 3D-Printable CAD Model Generation

After determining the exact arrangement of all logic gates and air channels, the final step is to gather all information and correspondingly generate a 3D-printable CAD file of a soft connection hub. The connection hub comes with embedded air channels that facilitate the interconnection of all individual gates, forming the complete pneumatic circuit. The connection hub comes with 1) attachment ports on the top to attach the logic gates according to the preplanned layout; 2) the pre-planned air channels for signal communication between logic gates; and 3) ports on the bottom as the total inputs/outputs of the entire computational circuit. The specific structure of the hub is shown in **Figure 4**. Its fabrication is feasible as long as the 3D printer has the capability to print both soluble support and airtight membranes. Once the CAD file of the soft connection hub is fabricated, the modular logic gates can then be attached directly to the hub without requiring any tools. The compliance of the connection hub and the press



**Figure 4.** The detailed hardware structure of the computational circuit. A) An exploded view of the circuit. A connection hub is used to connect all logic gates into the circuit. It includes some attachment ports for fitting with the logic gates on top, air channels in the middle, and a set of input/output ports that connect to the input and output pneumatic signal of the entire computation circuit. B) The side view of the final pneumatic circuit.



fit tolerance between the connection hub ports and the logic gates port ensure reliable fitting and prevent leakage.

Comsol Multiphysics via Livelink for MATLAB is used here to automatically generate such a 3D CAD file based on the preacquired information. Comsol is chosen due to its seamless integration with MATLAB, popularity in the soft robotics community, and the potential to be used for further finite element simulation. The toolbox first generates a layer with only ports located at the inputs and outputs of the entire circuit. It then proceeds to create the air channel layers as specified by the preacquired layout. Note that there could be more than one air channel layer, in which case different air channel layers will be isolated by dummy membranes. Finally, according to the preplanned logic gate layout, the algorithm generates a layer with ports located at the inputs and outputs of all logic gates. All generated layers are stacked together, forming the overall CAD file of the connection hub.

### 3. Results

#### 3.1. Validity of the Design Automation Algorithm

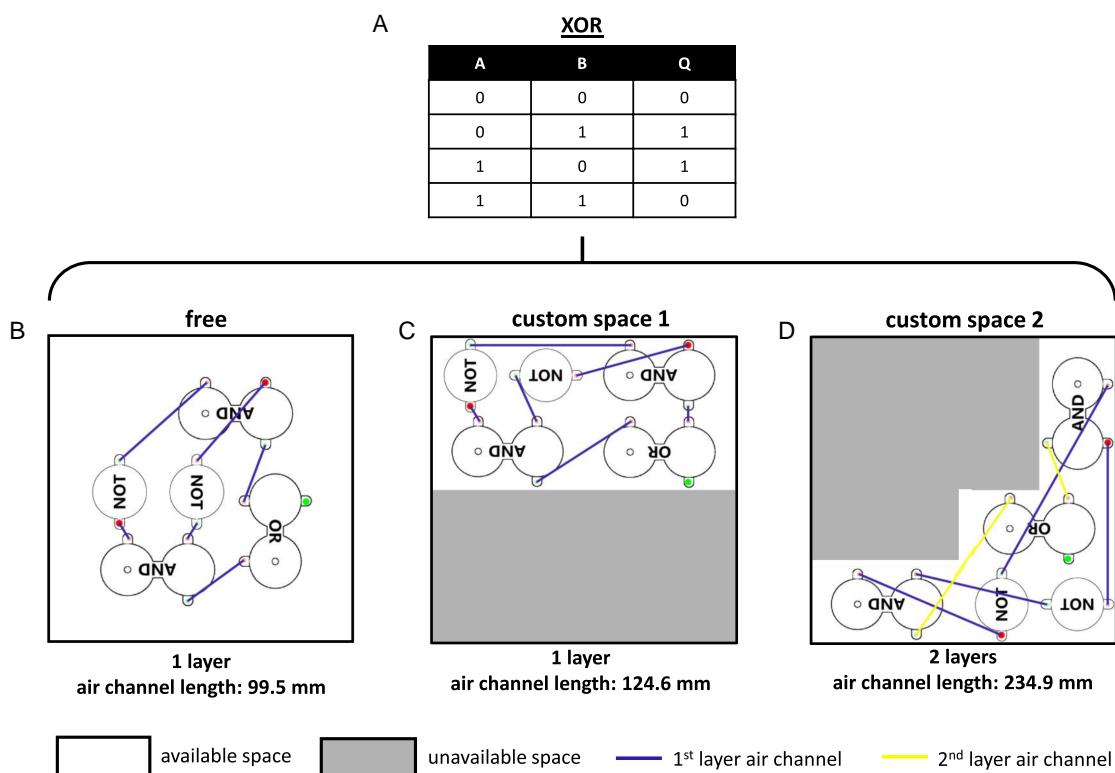
To explore the versatility of the design automation algorithm across different input–output mappings and space constraints, the truth tables for two commonly used digital computations, namely, XOR and 1:4 de-multiplexer, were inputted into the toolbox, alongside with different space constraints. XOR is chosen here due to its capability to perform various crucial logical

operations, facilitate error detection, and enable arithmetic operations. 1:4 demultiplexer is chosen here to demonstrate the compatibility of the toolbox algorithm when dealing with multiple–output truth tables. The pneumatic circuits planned by the algorithm to achieve XOR and 1:4 de-multiplexer within different 2D workspaces are presented in Figure 5 and 6, respectively.

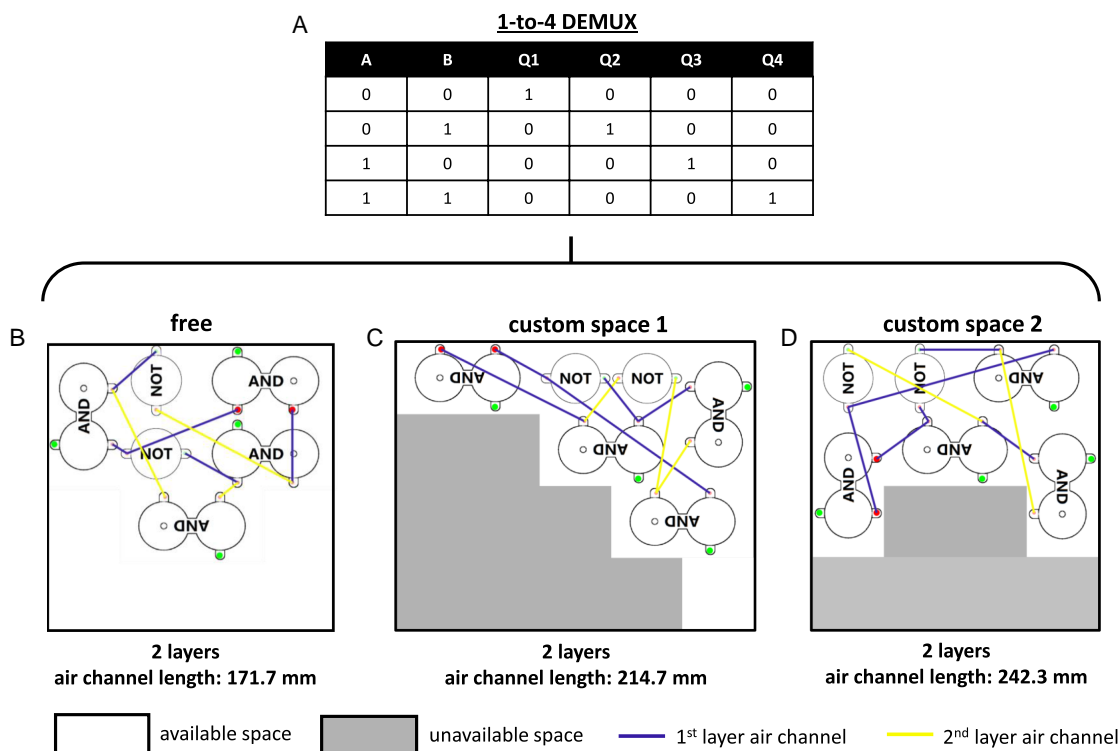
These results show that the algorithm is able to correctly compile the pneumatic circuits with a minimum number of logic gates while keeping all air channels and logic gates within the given workspace. All intersected pairs of air channels are correctly placed in different layers. It was also noted that for a given truth table, the total number of layers and the total length of air channels inevitably increase when more strict space constraints are engaged.

#### 3.2. Validity of the Physical Output

The autogenerated CAD files of the pneumatic circuits in Section 3.1 were fabricated by a polyjet multimaterial 3D printer (J735, Stratasys, USA). The performance of these circuits was then tested physically with respect to their computational correctness, response time, and long-term durability. Computational correctness describes whether the physical circuit is able to correctly implement the user-input truth table. This was tested by applying different pneumatic inputs into the circuits and monitoring the corresponding output signals. The response time is also crucial as the computation speed determines the



**Figure 5.** Pneumatic circuits under different space constraints scheduled by the toolbox to achieve an XOR computation. A) The user-input truth table taken by the toolbox. B–D) The different pneumatic circuits scheduled by the toolbox under different space constraints provided by users. The available workspace is indicated in white while the unavailable workspace is in gray. The color of each air channel indicates which layer it should be placed within. The larger red dots and green dots represent the inputs and outputs of the entire computational circuit, respectively.



**Figure 6.** Pneumatic circuits under different space constraints scheduled by the toolbox to achieve a 1:4 DEMUX computation. A) The user-input truth table taken by the toolbox. B–D) The different pneumatic circuits scheduled by the toolbox under different space constraints provided by users. The available workspace is indicated in white while the unavailable workspace is in gray. The color of each air channel indicates which layer it should be placed within. The larger red dots and green dots represent the inputs and outputs of the entire computational circuit, respectively.

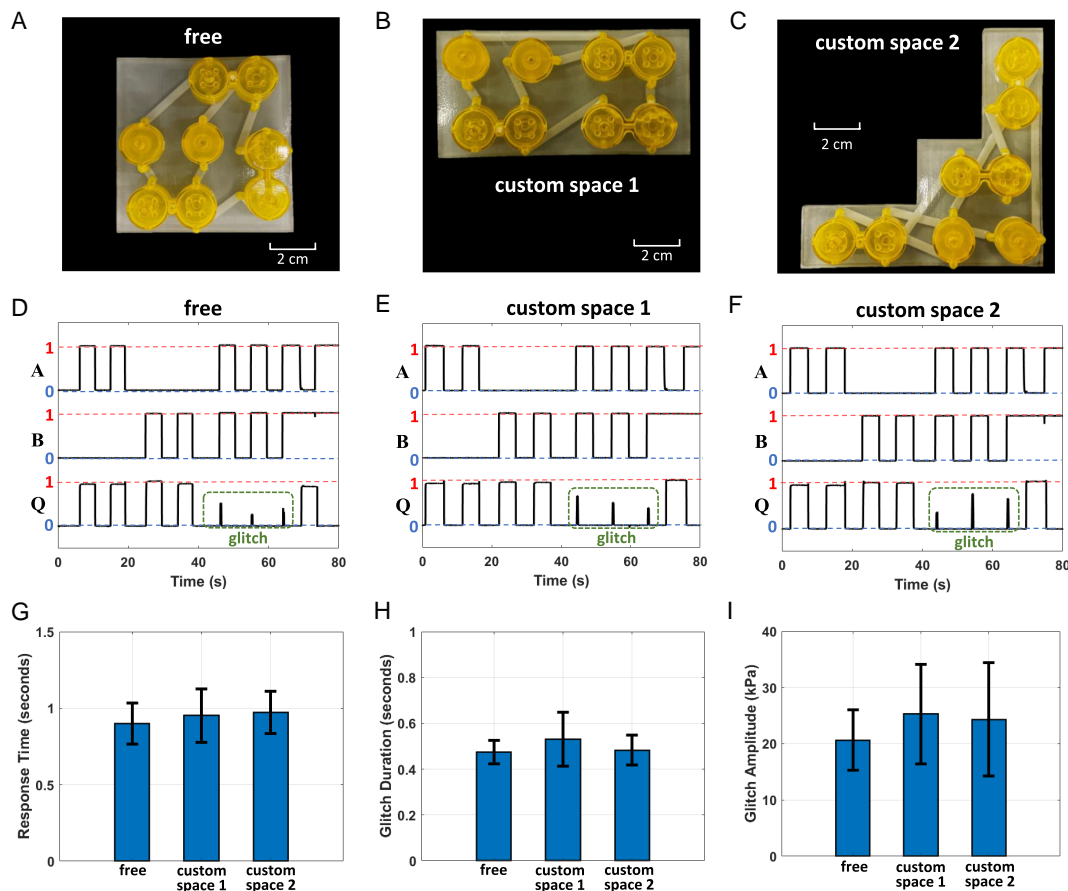
responsiveness and agility of the soft robots is controlled. It was measured as the time duration from the change in the input signal to the change observed in the output signals. Long-term durability under different operating frequencies was also investigated in terms of computation accuracy over repeated cycles. The experimental results of the physical XOR and 1:4 demultiplexer circuits with different space constraints are presented in Figure 7 and 8.

Figure 7A–C shows the three actual XOR circuits fabricated from the CAD files generated by our toolbox, as shown in Figure 5. The time-series input and output pressure readings in Figure 7D–F show that all these circuits correctly implement the desired truth table. The pneumatic circuits were tested with ten repeat trials while the recorded response time for each circuit is shown in Figure 7G. It can be seen that the pneumatic circuit scheduled without any space constraints has the fastest response time due to its shortest air channel lengths and least number of layers. “Glitches” were observed during the operation of all three XOR circuits in the form of unwanted spikes when both input signals are HIGH. The unwanted pneumatic spikes are caused by propagation delays when signals take different paths with different numbers of components before reaching the same output. The duration of the glitch in this XOR circuit should in theory be similar to the response time of a single NOT gate, which matches our measured value of 0.46 s, as shown in Figure 7H. The measured amplitude of the glitches is also presented in Figure 7I. The average glitch amplitude is measured as 22.6 kPa, which is 56.5% of the source pressure. In real soft

robotic applications, this glitch amplitude will be much smaller because the pneumatic chamber of the actuator acts as a capacitor which slows down the pressure rise. The compliant nature and the generally slow mechanical response of the soft pneumatic actuators also reduce the severity of the glitch. If precise timing is indeed required, the glitch can also be solved by adding repeaters or buffers to balance the signal propagation delay in each path. However, it should be noted that this delay-matching approach has not been addressed in the design automation algorithm described in this work.

Figure 8A–C shows the three actual 1:4 DEMUX circuits fabricated from the CAD files generated by our toolbox, as depicted in Figure 8. The time-series input and output pressure readings in Figure 8D–F indicate that all these circuits correctly implement the desired truth table. The pneumatic circuits were tested with ten repeat trials while the recorded response time of each output in each circuit is shown in Figure 8G. Since the 1:4 DEMUX circuit has multiple outputs, the response time of each individual output is recorded. The response time here is also measured as the time difference between the engagement of inputs and the full development of outputs. By comparing the overall performance across the three different pneumatic circuits, it can be found that the pneumatic circuit with the least air channel lengths comes with the fastest response. The shorter air channel means less energy and time is required to change its internal pressure, therefore speeding up the signal propagation along these channels and reducing the overall response time.





**Figure 7.** A–C) Three different fabricated pneumatic circuits to achieve XOR computation, scheduled by the toolbox as previously to suit different space constraints. D–F) The pressure–time curve of each pneumatic circuit at their inputs and outputs, showing the correct implementation of the desired truth table. G) The response time of each of the fabricated circuits. H) The duration of the undesired “glitch” observed in each circuit. I) The pressure amplitude of the “glitch” observed in each circuit.

One should also notice that the response time of different outputs in the same pneumatic circuit can differ significantly. In all three circuits, the output Q4 (only HIGH when both inputs are HIGH) has the lowest response time with an average value of 0.28 s, while the output Q1 (only HIGH when both inputs are LOW) comes with the highest response time of 0.82 s. This inconsistency is caused by the different logic gates located upstream of the different output ports. For instance, the output Q4 has NOT gates located in both of its upstream paths while the output Q1 has no NOT gates located upstream.

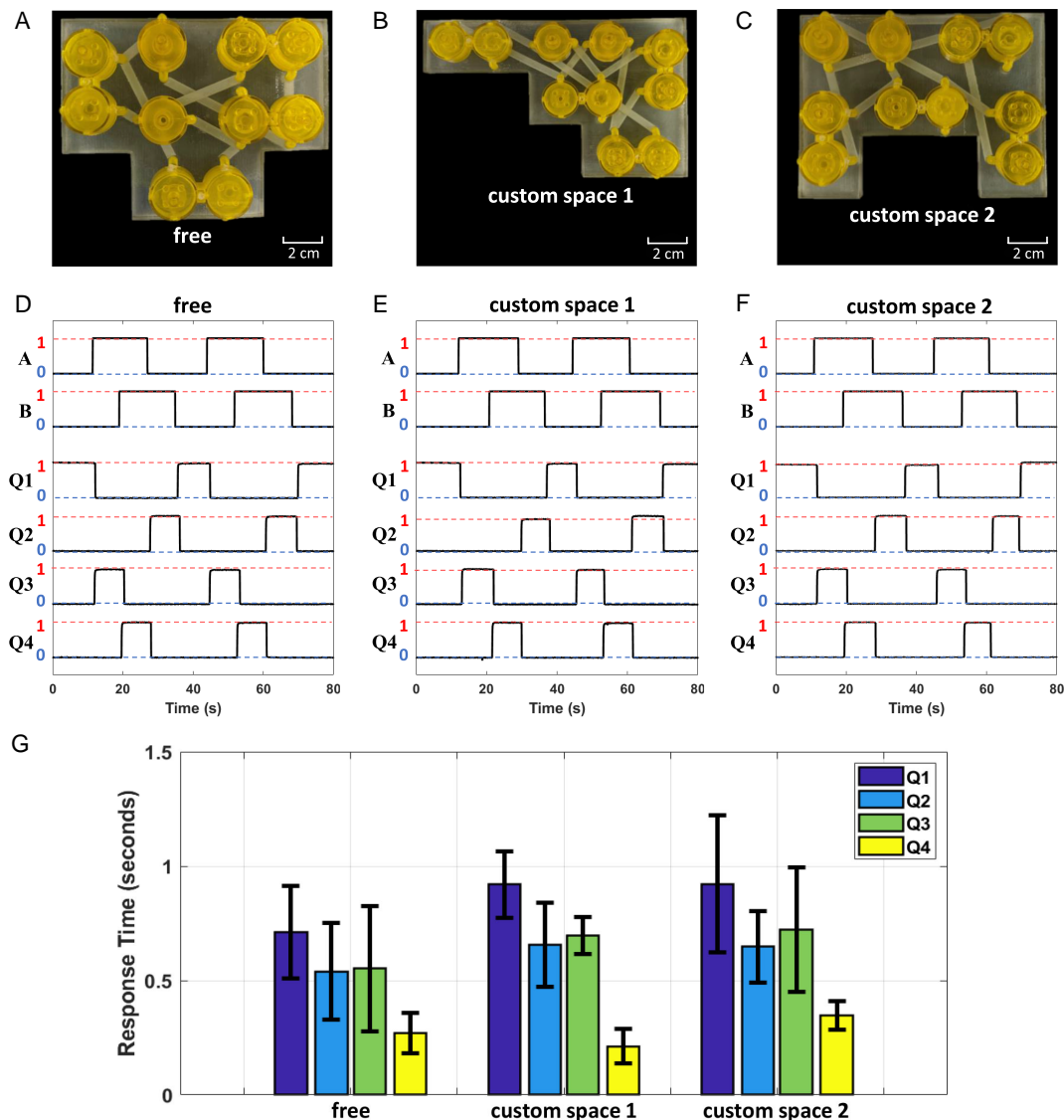
One should note that these pneumatic circuits are undoubtedly slower than the electric circuits. The pneumatic circuits with a response time of 0.5–1 s have been proven to be useful in various soft robotic tasks such as legged locomotion,<sup>[13]</sup> underwater locomotion,<sup>[22]</sup> grasping reflexes,<sup>[18]</sup> smart robotic wearables,<sup>[24]</sup> and more. However, these circuits may not be suitable for scenarios requiring fast response.

### 3.3. Reliability of the Printed Pneumatic Circuit

The long-term reliability of the pneumatic circuits was investigated through cycling tests, where different input signals were

supplied to the circuits repeatedly and the output pneumatic signals were recorded. The computation accuracy, defined as the number of correct computations in every 100 cycles, was plotted against the cycle number to reflect the durability of the pneumatic computation circuits. For each computation circuit, two different frequencies were used to perform the fatigue tests. With the lower frequency, the time duration of each input signal is 3 s, which is much higher than the response time of the circuit itself. The circuit should not fail under this frequency unless structural failure happens. With the higher frequency, the time duration for each input signal is 1.2 s, only slightly higher than the response time of the circuit itself. This test aims to find out how the responsiveness of the circuit evolves over usage since the circuit will fail under this frequency even with a small degradation in responsiveness. The fatigue test result is shown in **Figure 9**.

It can be seen that both the XOR and 1:4 DEMUX circuits successfully pass through 10 000 cycles with 100% computation accuracy under the lower frequency (signal duration  $T = 3$  s). No structural failure was observed after the fatigue tests. For the fatigue tests with higher frequency, it can be seen that the responsiveness of both circuits degrades over usage.



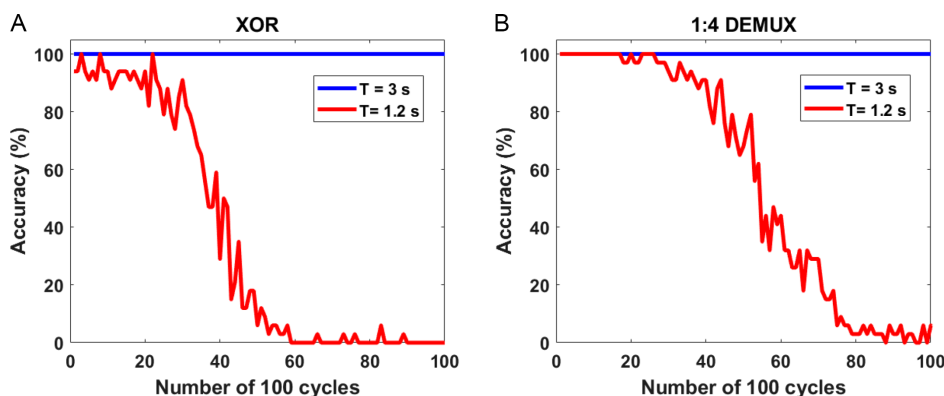
**Figure 8.** A–C) The three different fabricated pneumatic circuits to achieve the 1:4 demultiplexer, scheduled by the toolbox as previously to suit different space constraints. D–F) The pressure–time curve of each pneumatic circuit at their inputs and outputs, showing the correct implementation of the desired truth table. G) The response time of the four outputs measured from the different pneumatic circuits.

The computation accuracy of the XOR circuit drops below 10% after around 5000 cycles, indicating that the response time of the circuit is longer than the input signal duration of 1.2 s. The 1:4 DEMUX circuit has a slower degradation in response time, compared with the XOR circuit. Its computation accuracy under high frequency drops below 10% after 8000 cycles. The degradation in response time is mainly caused by the fatigue and aging of the soft material located in each logic gate. The degradation in response time is mainly caused by the fatigue and aging of the soft material located in each logic gate. Agilus 30 is an acrylic-based polyjet photopolymer that tends to increase its stiffness and viscoelasticity over long-term operation.<sup>[34]</sup> This aging increases the response time of the logic gates when the input pneumatic pressure is kept constant. This degradation in response time should be carefully considered when applying

these circuits in long-term and high-frequency computation scenarios.

### 3.4. Four-Finger Robotic Hand Controlled by Pneumatic Circuit

To showcase the application of the automated toolbox when designing pneumatic circuits for soft robots, we present a soft robotic hand with four pneumatic fingers whose movements are controlled by a pneumatic circuit. The four fingers are made with PneuNet actuators,<sup>[39]</sup> while the pneumatic circuit is located in the palm. The truth table of a 1:4 demultiplexer and the shape of the palm are then input into the design automation toolbox. The generated pneumatic circuit was then applied to the palm of the soft robotic hand, while the four output ports were connected to the four pneumatic fingers. The different configurations of the



**Figure 9.** Results of the fatigue tests. A,B) Long-term computation accuracy of the XOR and 1:4 DEMUX circuit, respectively. The tests were performed with different input signal durations to investigate the circuits' durability under different computation frequencies. Each data point in the plots refers to the number of correct computations within 100 cycles.

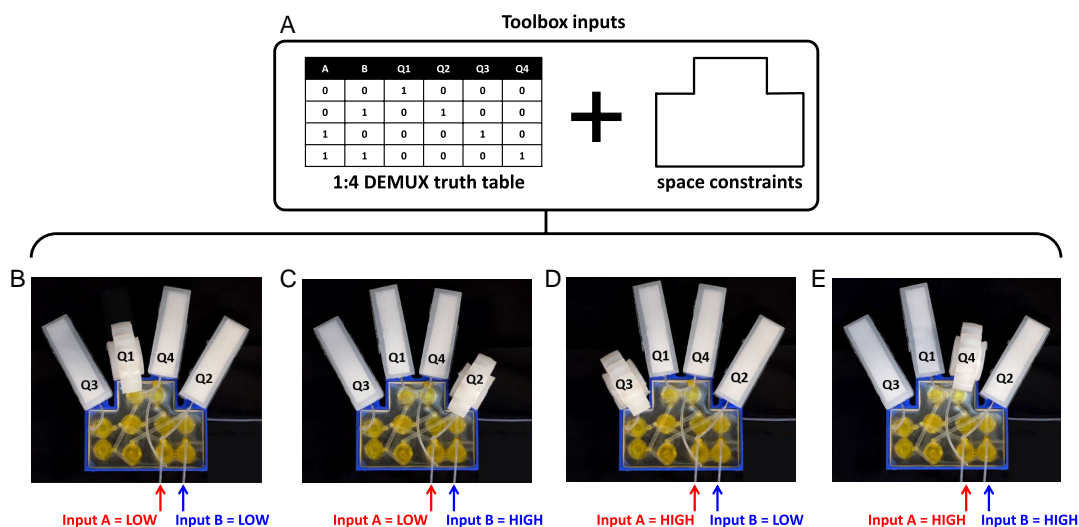
soft robotic hand under different input signals are presented in **Figure 10**. It can be seen that the pneumatic circuit correctly implements the desired input–output mapping and correspondingly controls the four pneumatic fingers onboard.

## 4. Conclusion

This work presents a design automation toolbox that generates a 3D-printable computational pneumatic circuit to achieve a desired truth table while adhering to certain space constraints. Unlike the previous work, which solely converts user-input truth tables into conceptual circuit diagrams, this study goes a step further by accomplishing the transition from software-based circuit schematics to their physical copies. This is achieved by accurately positioning logic gates and air channels, taking into account workspace constraints, fabrication limits, material cost, and system complexity. The toolbox has the capability to replace the manual circuit design, fabrication, and piping process that used

to be highly challenging and time-consuming. The automated approach also eliminates redundant tubing and unreliable fitting, therefore enhancing the responsiveness and durability of the computational circuit.

There are still various avenues for future development that can build upon the current toolbox architecture. First, the current toolbox chooses the optimal circuit layout only based on the minimum number of layers and least channel lengths. More diverse cost functions can be introduced in the future, such as Center of Mass (CoM) positioning, input–output port positioning, and so on. Second, the current optimization process assumes the ideal operation of all logic gates without considering their nonlinearities, different pressure losses, or different response times. This brings problems such as inconsistent response time, signal glitches, or even computation mistakes when the circuit gets more complex. It is worth including the nonidealities of these gates in the design automation algorithms by modeling the response time of different logic gates, the fluidic dynamics in air channels, and the pressure loss across each circuit



**Figure 10.** A four-finger soft robotic hand controlled by a 1:4 DEMUX pneumatic circuit. A) The input–output mapping and space constraints fed into the design automation toolbox. B–E) Configurations of the soft robotic hand with different input signals.

component. Including these factors in the design automation algorithms enables more complex pneumatic circuits to be built with more reliable outputs, more consistent response time, and less energy consumption. Third, the current toolbox only includes the 3D-printed logic gates in its library, limiting the computation capability to simple logic mapping. Enriching its library with other soft pneumatic devices can extend the pneumatic circuits to more advanced computation and better fault tolerance, therefore achieving more precise and robust control of soft robots.<sup>[25,40]</sup> Finally, the current toolbox schedules the circuits in a 2D manner, where all logic gates are located in the same plane. This brings limitations when applied to more complex and large-scale systems. 3D circuit scheduling could be addressed in the future to further broaden its applications in soft robots.

## 5. Experimental Section

**Detailed Description of the Toolbox GUI:** The installation and setup of the toolbox software can be found in Supporting Information S1. Once installed and correctly set up, the toolbox comes with a GUI, as shown in Figure 1. The following procedures are implemented when using this toolbox for designing pneumatic computational circuits for soft robotic applications: 1) Define input/output number: The first step is to tell the toolbox how many inputs and outputs are included in your desired truth table. The toolbox will automatically generate a truth table below based on these numbers; 2) Edit the truth table: The default truth table generated by the toolbox has LOW output at all times. This needs to be modified to suit the user's desired input–output mapping by directly editing the table. Any nonzero value in the cell will be treated as a HIGH signal, while zero will be treated as a LOW signal; and 3) Define the available 2D space: The space constraints should be input through the editable 2D map. Available space is represented by white, while the unavailable space is represented by dark grey. The availability of each unit can be switched by direct clicking, therefore enabling the user to customize the available workspace to suit different applications; 4) Plan the pneumatic circuit: Once the input–output mapping and space constraints are all set, the user can click the “Compile the circuit!” button. The design automation algorithm will then schedule the pneumatic computational circuit to achieve the desired truth table within the space constraints while minimizing the number of air channel layers and air channel lengths. The generated circuit schedule will then be displayed in the user interface, indicating the placements of all logic gates and the air channels. The color of each air channel indicates which layer it should be placed within. The larger red dots and green dots represent the inputs and outputs of the entire computational circuit, respectively; and 5) Generate the ready-to-print CAD files: After scheduling the pneumatic circuit, the user can generate the STereoLithography (STL) files for all logic gates and the connection hub with embedded air channels by pressing the “Save STL” button. Pressing the button opens a file explorer where you can specify the path where the STL files should be saved.

**Fabrication of the Pneumatic Circuit:** The pneumatic computational circuit was printed by a Polyjet-based multimaterial 3D printer (J735, Stratasys, USA). The printer setup was implemented in its corresponding slicer software (GrabCAD, Stratasys, USA). The connection hub of the pneumatic circuit was set to be printed with Agilus 30 (a rubber-like material with a quoted tensile strength of 2.1–2.6 MPa and Shore hardness of 30A). The compliant part in each logic gate, shown as black in Figure 2, was set to be printed with Agilus30 as well. The rigid part in each logic gate, shown as white in Figure 2, was set to be printed with VeroYellow Vivid (a rigid plastic-like material with a quoted tensile strength of 50–65 MPa and Shore hardness of 83–86D). The model finish was set to “Glossy”, while the support strength was set to “Lite.” Once the printing finished, the pneumatic circuit was placed in a tank (GEMINI SSR-550)

filled with chemical support removal solution (0.02 kg L<sup>−1</sup> Sodium Hydroxide and 0.01 kg L<sup>−1</sup> Sodium Metasilicate) for support removal.

**Experimental Setup for Testing the Pneumatic Circuits:** To apply different input signals to the pneumatic computational circuits, a pneumatic system was used. The air was first compressed by an oil-free scroll compressor (SLPE-07ED, ANEST IWATA, Japan). A pneumatic regulator (AR20-F02-1-B, SMC, Japan) was then used to regulate the pneumatic pressure. Solenoid valves (VDW10AA, SMC, Japan) were used to control the engagement of the compressed air to the 3D-printed pneumatic circuits. Analog pressure sensors (ADP51A11, Panasonic, Japan) were used to monitor the pneumatic signals output by the computational circuits. All sensory data were captured through Teensy 4.1, a development platform that features an ARM Cortex-M7 processor at 600 MHz.

## Supporting Information

Supporting Information is available from the Wiley Online Library or from the author.

## Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) grant EP/V000748/1. S.W. was supported by the CSC-PAG Oxford Scholarship for his D.Phil. study.

## Conflict of Interest

The authors declare no conflict of interest.

## Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

## Keywords

3D printing, design automation, fluidic circuits, pneumatic actuators, soft robotics

Received: July 7, 2023  
Revised: September 17, 2023  
Published online: October 18, 2023

- [1] G. M. Whitesides, *Angew. Chem., Int. Ed.* **2018**, *57*, 4258.
- [2] D. Drotman, S. Jadhav, M. Karimi, P. de Zonia, M. T. Tolley, in *2017 IEEE Int. Conf. on Robotics and Automation (ICRA)*, IEEE, Piscataway, NJ **2017**, pp. 5532–5538.
- [3] R. Das, S. P. M. Babu, F. Visentin, S. Palagi, B. Mazzolai, *Sci. Rep.* **2023**, *13*, 1571.
- [4] O. Shorthose, A. Albini, L. He, P. Maiolino, *IEEE Rob. Autom. Lett.* **2022**, *7*, 3945.
- [5] S. Jain, S. Dontu, J. E. M. Teoh, P. Valdivia, Y. Alvarado, *Soft Rob.* **2023**, *10*, 527.
- [6] J. D. Greer, L. H. Blumenschein, A. M. Okamura, E. W. Hawkes, in *2018 IEEE Int. Conf. on Robotics and Automation (ICRA)*, IEEE, Piscataway, NJ **2018**, pp. 4165–4172.
- [7] J. Walker, T. Zidek, C. Harbel, S. Yoon, F. S. Strickland, S. Kumar, M. Shin, *Actuators* **2020**, *9*, 3.
- [8] Y. Tang, Y. Chi, J. Sun, T.-H. Huang, O. H. Maghsoudi, A. Spence, J. Zhao, H. Su, J. Yin, *Sci. Adv.* **2020**, *6*, eaaz6912.

- [9] C. M. Tan, V. K. Pandey, Y. Chiang, T. P. Lee, *Sens. Mater.* **2022**, *34*, 1119.
- [10] A. Roguin, J. Schwitter, C. Vahlhaus, M. Lombardi, J. Brugada, P. Vardas, A. Auricchio, S. Priori, T. Sommer, *Europace* **2008**, *10*, 336.
- [11] C. D. Onal, X. Chen, G. M. Whitesides, D. Rus, in *Robotics Research: The 15th Int. Symp. ISRR*, Springer, New York, NY **2017**, pp. 525–540.
- [12] M. Wehner, R. L. Truby, D. J. Fitzgerald, B. Mosadegh, G. M. Whitesides, J. A. Lewis, R. J. Wood, *Nature* **2016**, *536*, 451.
- [13] D. Drotman, S. Jadhav, D. Sharp, C. Chan, M. T. Tolley, *Sci. Rob.* **2021**, *6*, eaay2627.
- [14] L. C. van Laake, J. de Vries, S. M. Kani, J. T. Overvelde, *Matter* **2022**, *5*, 2898.
- [15] K. McDonald, T. Ranzani, *Front. Rob. AI* **2021**, 266.
- [16] Y. Masuda, M. Ishikawa, *J. Rob. Mechatron.* **2022**, *34*, 202.
- [17] Q. Lu, H. Xu, Y. Guo, J. Y. Wang, L. Yao, in *Proc. of the 2023 CHI Conf. on Human Factors in Computing Systems*, Association for Computing Machinery, New York, NY **2023**, pp. 1–21.
- [18] P. Rothmund, A. Ainla, L. Belding, D. J. Preston, S. Kurihara, Z. Suo, G. M. Whitesides, *Sci. Rob.* **2018**, *3*, eaar7986.
- [19] D. J. Preston, P. Rothmund, H. J. Jiang, M. P. Nemitz, J. Rawson, Z. Suo, G. M. Whitesides, *Proc. Natl. Acad. Sci. USA* **2019**, *116*, 7750.
- [20] D. J. Preston, H. J. Jiang, V. Sanchez, P. Rothmund, J. Rawson, M. P. Nemitz, W.-K. Lee, Z. Suo, C. J. Walsh, G. M. Whitesides, *Sci. Rob.* **2019**, *4*, eaaw5496.
- [21] M. P. Nemitz, C. K. Abrahamsson, L. Wille, A. A. Stokes, D. J. Preston, G. M. Whitesides, in *2020 3rd IEEE Int. Conf. on Soft Robotics (RoboSoft)*, IEEE, Piscataway, NJ **2020**, pp. 7–12.
- [22] K. Bonofiglio, L. Whiteside, M. Angeles, M. Haahr, B. Simpson, J. Palmer, Y. Wu, M. P. Nemitz, in *2023 IEEE Int. Conf. on Soft Robotics (RoboSoft)*, IEEE, Piscataway, NJ **2023**, pp. 1–6.
- [23] J. A. Tracz, L. Wille, D. Pathiraja, S. V. Kendre, R. Pfisterer, E. Turett, C. K. Abrahamsson, S. E. Root, W.-K. Lee, D. J. Preston, H. J. Jiang, G. M. Whitesides, M. P. Nemitz, *IEEE Rob. Autom. Lett.* **2022**, *7*, 5483.
- [24] A. Rajappan, B. Jumet, R. A. Shveda, C. J. Decker, Z. Liu, T. F. Yap, V. Sanchez, D. J. Preston, *Proc. Natl. Acad. Sci. USA* **2022**, *119*, e2202118119.
- [25] C. J. Decker, H. J. Jiang, M. P. Nemitz, S. E. Root, A. Rajappan, J. T. Alvarez, J. Tracz, L. Wille, D. J. Preston, G. M. Whitesides, *Proc. Natl. Acad. Sci. USA* **2022**, *119*, e2205922119.
- [26] G. Huang, J. Hu, Y. He, J. Liu, M. Ma, Z. Shen, J. Wu, Y. Xu, H. Zhang, K. Zhong, X. Ning, Y. Ma, H. Yang, B. Yu, H. Yang, Y. Wang, *ACM Trans. Des. Autom. Electron. Syst.* **2021**, *26*, 40.
- [27] D. Grissom, C. Curtis, S. Windh, C. Phung, N. Kumar, Z. Zimmerman, O. Kenneth, J. McDaniel, N. Liao, P. Brisk, *Integr. VLSI J.* **2015**, *51*, 169.
- [28] A. A. Nielsen, B. S. Der, J. Shin, P. Vaidyanathan, V. Paralanov, E. A. Strychalski, D. Ross, D. Densmore, C. A. Voigt, *Science* **2016**, *352*, aac7341.
- [29] SMC UK, Pneumatic Circuits Drawing, [https://www.smc.eu/en-gb/products/engineering-tools/pneumatic\\_circuits\\_drawing](https://www.smc.eu/en-gb/products/engineering-tools/pneumatic_circuits_drawing) (accessed: 2023).
- [30] Festo, Fluiddraw p6, <https://www.festo.com/gb/en/app/fluiddraw-p6.html> (accessed: 2023).
- [31] Festo, Fluidsim, [https://www.festo.com/ae/en/e/technical-education/digital-learning/virtual-simulation-and-modelling/fluidsim-id\\_1663056/](https://www.festo.com/ae/en/e/technical-education/digital-learning/virtual-simulation-and-modelling/fluidsim-id_1663056/) (accessed: 2023).
- [32] S. V. Kendre, L. Whiteside, T. Y. Fan, J. A. Tracz, G. T. Teran, T. C. Underwood, M. E. Sayed, H. J. Jiang, A. A. Stokes, D. J. Preston, G. M. Whitesides, M. P. Nemitz, *IEEE Rob. Autom. Lett.* **2022**, *7*, 6060.
- [33] J. D. Hubbard, R. Acevedo, K. M. Edwards, A. T. Alsharhan, Z. Wen, J. Landry, K. Wang, S. Schaffer, R. D. Sochol, *Sci. Adv.* **2021**, *7*, eabe5257.
- [34] S. Wang, L. He, P. Maiolino, *IEEE Rob. Autom. Lett.* **2021**, *7*, 112.
- [35] S. Wang, L. He, P. Maiolino, *IEEE Rob. Autom. Lett.* **2022**, *7*, 3412.
- [36] P. Codognet, F. Fages, T. Sola, in *Constraint Logic Programming, Selected Research. WCLP 1991, Marseilles, France*, MIT Press, Pennsylvania, United States **1993**, pp. 437–456.
- [37] A. H. Ashouri, A. Bignoli, G. Palermo, C. Silvano, in *Proc. of the 7th Workshop on Parallel Programming and Run-Time Management Techniques for Many-Core Architectures and the 5th Workshop on Design Tools and Architectures for Multicore Embedded Computing Platforms*, Association for Computing Machinery, New York, NY **2016**, pp. 7–12.
- [38] J. Pach, M. Sharir, *SIAM J. Comput.* **1991**, *20*, 460.
- [39] B. Mosadegh, P. Polygerinos, C. Keplinger, S. Wennstedt, R. F. Shepherd, U. Gupta, J. Shim, K. Bertoldi, C. J. Walsh, G. M. Whitesides, *Adv. Funct. Mater.* **2014**, *24*, 2163.
- [40] M. Pontin, S. Miyashita, D. D. Damian, in *2022 IEEE 5th Int. Conf. on Soft Robotics (RoboSoft)*, IEEE, Piscataway, NJ **2022**, pp. 62–67.