Formal Synthesis of Adaptive Droplet Routing for MEDA Biochips[*]

Mahmoud Elfar, Tung-Che Liang, Krishnendu Chakrabarty, and Miroslav Pajic

***Abstract*—A digital microfluidic biochip (DMFB) enables the miniaturization of immunoassays, point-of-care clinical diagnostics, and DNA sequencing. A recent generation of DMFBs uses a micro-electrode-dot-array (MEDA) architecture, which provides fine-grained control of droplets and real-time droplet sensing using CMOS technology. However, microelectrodes in a MEDA biochip can degrade due to charge trapping when they are repeatedly charged and discharged during bioassay execution; such degradation leads to the failure of microelectrodes and erroneous bioassay outcomes. To address this problem, we first introduce a new microelectrode-cell design such that we can obtain the health status of all the microelectrodes in a MEDA biochip by employing the inherent sensing mechanism. Next, we present a stochastic game-based model for droplet manipulation, and a formal synthesis method for droplet routing that can dynamically change droplet transportation routes. This adaptation is based on the real-time health information obtained from microelectrodes. Comprehensive simulation results for four real-life bioassays show that our method increases the likelihood of successful bioassay completion with negligible impact on time-to-results.**

I. Introduction

Digital microfluidic biochips (DMFBs) are being adopted for biomolecular recognition, point-of-care diagnostics, and air-quality monitoring applications [1], [2], [3]. A microfluidic biochip manipulates tiny amounts of fluids to automatically execute biochemical protocols for point-of-care clinical diagnosis with high efficiency and fast sample-to-result turnaround [4], [5]. Because of these characteristics, the Rapid Acceleration of Diagnostics (RADx) initiative from the National Institutes of Health has recently awarded grants to several biomedical diagnostic companies to develop microfluidic technologies that can dramatically increase COVID-19 testing capacity and throughput [6]. This technology has been commercialized in recent years for clinical diagnostics and immunoassays [7], [8].

Micro-electrode-dot-array (MEDA) biochips have been proposed in recent years to further advance DMFB technology [9]. A MEDA biochip manipulates fluids as discrete droplets of picoliter volume using the principle of electro-wetting-on-dielectric (EWOD) [10]. In addition, MEDA biochips also offer fine-grained fluidic control and real-time droplet sensing on a two-dimensional array of microelectrodes [10], [9]. Multiple microelectrodes can be dynamically grouped to form a fluidic module (e.g., splitter or mixer). MEDA biochips have been fabricated using TSMC $0.35\,\mu\mathrm{m}$ CMOS technology [11].

In the MEDA platform, a real-time capacitive-sensing circuit is integrated with each microelectrode to detect the location and properties of a droplet. In each operational cycle, the sensing circuit discharges and charges the microelectrode, and measures the charging time. The charging time is used to detect whether a droplet is present over the microelectrode. To obtain the positions of on-chip droplets, the sensing results of all the microelectrodes are shifted out using a scan chain.

Prior work has identified a number of failure mechanisms for DMFBs [12]. Some of these are related to manufacturing defects; post-fabrication testing can be used to screen for such defects [13]. However, charge trapping in the dielectric layer and degradation of the insulator can result in microelectrode degradation [14], [15]. If an electrode is degraded during bioassay execution, fluidic operations associated with this degraded electrode will fail, resulting in bioassay failure [16], [17]. Hence, to reliably execute bioassays on MEDA biochips, we must proactively avoid the use of degraded microelectrodes. The majority of literature on reliable execution of bioassays on both DMFBs and MEDA biochips has focused on error recovery techniques. Fundamentally, error recovery techniques aim to reactively recover from errors in microfluidic operations after they occur. However, error recovery techniques may require discarding current droplets and repeating a number of microfluidic operations, potentially losing expensive and/or hard to reproduce droplets.

In this paper, we address the problem of bioassay failure due to microelectrode degradation by proposing an adaptive routing strategy synthesis framework. We first present a new microelectrode circuit design that provides dynamic health information about the microelectrode degradation level in real time. Next, we develop a stochastic game-based model for MEDA biochips that incorporates the dynamic health information. The developed model is further used to induce Markov decision processes (MDPs) to present a scalable formal synthesis method for droplet routing that can dynamically change droplet transportation routes based on the real-time health information obtained. The main contributions of this paper are as follows.

- We introduce a MEDA biochip microelectrode design that enables the sensing of microelectrode health level in real time.
- We study how the microelectrode degradation patterns are affected by droplet characteristics for various bioassays. We also present experimental results on electrode degradation for fabricated PCB prototypes. These results are

[*] This research was supported in part by the National Science Foundation under grant No. ECCS-1914796.

Mahmoud Elfar, Tung-Che Liang, Krishnendu Chakrabarty, and Miroslav Pajic are with the Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA (e-mail: mahmoud.elfar@duke.edu; tung.che.liang@duke.edu; krish@duke.edu; miroslav.pajic@duke.edu).

used to validate the degradation model used throughout the paper.

- We develop a stochastic game-based model for droplet manipulation in MEDA biochips that incorporates the health information obtained from the new proposed design.
- We propose a formal synthesis framework where MDPs induced from the developed model are employed in the automatic synthesis of adaptive droplet routing strategies that maximize the likelihood of successful bioassay execution by adapting to the microelectrode health information in real time.
- We develop a MEDA biochip simulator based on the SMG model to evaluate the proposed framework and present simulation results for six benchmark bioassays.

The rest of this paper is organized as follows. Section II provides the notation used in this paper, as well as the necessary background on bioprotocol synthesis on DMFBs, microelectrode degradation, and error recovery techniques for biochips. In Section III, we present the new proposed microelectrode design and further explore degradation patterns in MEDA biochips. We also present experimental results on electrode degradation for fabricated PCB protoypes. Section IV describes both the experimental and theoretical relationship between microelectrode degradation and the number of actuations. Next, we develop a stochastic game-based model for MEDA biochips in Section V. Based on the developed model, we propose an adaptive routing strategy synthesis framework in Section VI. The experimental evaluation of the proposed framework is presented in Section VII. Finally, we draw our conclusions in Section VIII.

II. Background

A. Notation

$\mathbb{N}_0$ denotes the set of non-negative integers. For $a, b \in \mathbb{N}_0$, $[\![a, b]\!]$ denotes the discrete interval $\{x \mid x \in \mathbb{N}_0, a \leq x \leq b\}$. For a variable x, $x^{(k)}$ denotes its value at time $k \in \mathbb{N}_0$. We use bold symbols for matrices, e.g., $\mathbf{M} = (M_{ij}) \in \mathbb{R}^{m \times n}$; here, M_{ij} is the element in the i-th row and j-th column. The i-th element of a tuple or list w is denoted by $w[i-1]$, $i \in \mathbb{N}_0$. For a set A, $\mathcal{P}(A)$ is its power set. For a random variable x, $x \sim U(x_1, x_2)$ denotes that x follows a uniform distribution with the interval $[x_1, x_2] \subset \mathbb{R}$. We use N, S, E and W to denote north, south, east and west, respectively.

B. Bioprotocol Synthesis on DMFBs

In the DMFB synthesis flow [18], a bioassay protocol with specified fluidic operations is first developed by biologists. Next, a synthesis tool maps fluidic operations to fluidic modules on the electrode array [19], [20]. Droplets need to be transported as part of the bioprotocol synthesized on the DMFB. Various droplet-routing methods have been proposed in the literature [21], [22], [23], [24], including techniques that are specific to MEDA [25], [16]. However, these methods do not consider electrode degradation. Recently, reinforcement learning has been proposed to transport droplets in a reliable manner [17]. However, this approach does not monitor the dynamic health condition of electrodes and therefore fluidic operations associated with degraded electrodes may still fail.

C. Error Recovery Techniques

For both DMFBs and MEDA biochips, the goal of error recovery techniques is to detect and further correct errors that occur during a bioassay execution [26], [27]. Techniques in literature can be categorized based on the type of corrective action into two groups: retrial and roll-back. In retrial-based recovery techniques, attempts are carried out to correct errors detected in a microfluidic operation without discarding the droplets involved. Depending on the error type, the droplets involved might get rerouted, reshaped, re-mixed, resplit, or undergo a combination of these corrective actions.

III. Microelectrode Cell Design

A. MEDA Biochips and Microelectrode Cell

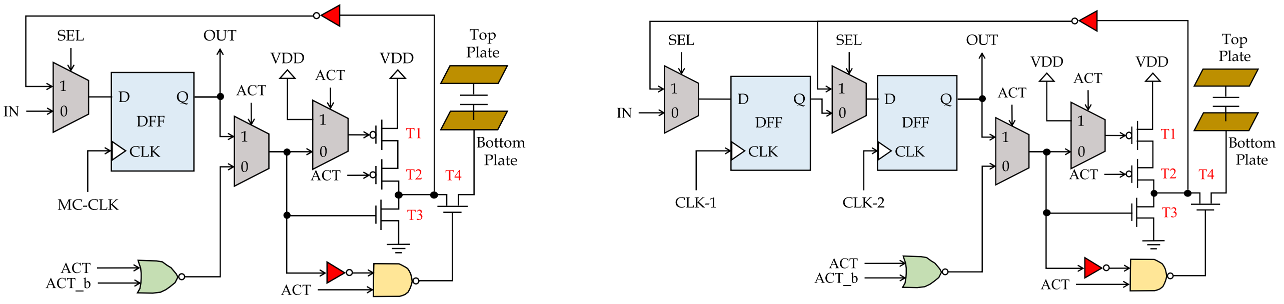
A MEDA biochip is composed of an array of identical microelectrode cells (MCs) and a controller; the schematic of an MC is shown in Fig. 1(a). Each MC consists of a microelectrode, an electronic control circuit, and a sensing module. To carry out a bioassay on a MEDA biochip, a synthesis tool is first used to generate a schedule of fluidic operations, module placement, and droplet routes for the bioassay [16]. These are next mapped to a sequence of actuation patterns. The actuation patterns are sequentially shifted to the MC array through a scan chain. The MCs are actuated based on the scanned-in data, and the corresponding fluidic operations are carried out based on EWOD. After MC actuation, all the MCs are set to the sensing mode to capture droplet locations. The sensing results are then scanned out as a bitstream. The process of shifting an actuation bitstream, MC actuation, droplet sensing, and shifting the sensing results is referred to as an *operational cycle*.

B. Microelectrode Degradation and Health Monitoring

MC sensing is used to detect droplet locations by measuring the capacitance between the top plate and bottom plate. The controller sets ACT = 0, ACT_b = 1, and SEL = 1; it also connects the top plate to ground. When this happens, transistors T1, T2, and T4 are switched on while transistor T3 is switched off, the bottom plate is connected to VDD (3.3 V) and the voltage of the bottom plate increases to 3.3 V. Next, the control circuit set ACT_b = 0, and transistors T1, T3 and T4 are switched on while transistor T2 is switched off. As a result, the bottom plate is now connected to ground, and the voltage of the bottom plate decreases due to discharging. By applying a rising edge of MC-CLK at a preset time, a value of “0” or “1” can be stored in the DFF.

A major contributor to microelectrode degradation is the gradual trapping of charge in the dielectric layer [15], [28], [29]. Thus, a proactive approach to ensure reliable fluidic operations is to estimate the degradation status of all microelectrodes in real-time and utilize only the healthy ones. To achieve this, we introduce a new MC design (Fig. 1(b)).

Charge trapping in the dielectric layer results in a higher capacitance between a degraded microelectrode and the top



(a)

(b)

Fig. 1: Schematics of (a) the original and (b) the new proposed microelectrode circuit models for MEDA biochips.

plate [30]; therefore, we can use capacitive sensing to detect degradation. An extra D flip-flop (DFF) is added to the MC design, and the rising edge of the CLK signal for this DFF is designed to arrive later than that of the other (original) DFF. For a healthy microelectrode, the 2-bit sensing result is “11”. If a microelectrode is partially degraded, the charging/discharging time is slightly less than that of a healthy microelectrode, and the original DFF is able to capture this difference by registering a different value compared to the newly added DFF (“0” versus “1”). If a microelectrode is completely degraded, the charging/discharging time is significantly lower than that of a healthy microelectrode, and both DFFs record “0”. This dynamic 2-bit sensing result provides the health-status information for the formal analysis model and synthesis method described in Sections V–VI.

We simulated the new MC design in HSPICE, using the macro-model for the extended-drain MOS transistors in the MC and a 350 nm library from a foundry; these models and parameters match the characteristics of fabricated biochips. We calculated the capacitance of microelectrodes using the parameters listed in Table I. The simulation results are shown in Fig. 2, where the rising edge of the clock signal of the added DFF needs to be asserted 5 ns later than that of the original DFF. Note that MCs are fabricated using CMOS technology and CMOS-based frequency dividers in the range of GHz are available [31]. Hence, by carefully controlling the rising edges of the two DFFs, we can dynamically measure the health status of a microelectrode. The added DFF has no impact on the chip footprint because its area ($\sim$27 μm^2) is much less than the area of a microelectrode (2,500 μm^2) minus the area of the electronics underneath it ($\sim$88.2 μm^2) [9]; the microfluidics part clearly dominates the overall area of the MC.

C. Degradation Patterns

In this subsection we examine the actuation patterns for which MC faults appear due to degradation. To this end, we design a set of experiments where we study the correlation between the number of actuations for two MCs and the Manhattan distance between them. In this set of experiments, we simulated the execution of three bioassays: ChIP, multiplex in vitro, and gene expression [32], on a 60×30 MEDA biochip. For each execution, the actuations of an MC at location (i, j) is recorded as a Boolean vector $A_{ij} \in \{0, 1\}^N$, where N is the number of operational cycles. The correlation coefficient between two MCs at locations (i, j) and (k, l) is defined as

$$\rho_{A_{ij},A_{kl}} = \frac{cov\left(A_{ij}, A_{kl}\right)}{\sigma_{A_{ij}}\sigma_{A_{kl}}},$$

where $cov\left(A_{ij}, A_{kl}\right)$ is the covariance between the two vectors, and $\sigma_{A_{ij}}$ is the standard deviation of A_{ij}.

TABLE I: Notation used for the simulation.

Symbols	Description	Values
A	Area of a microelectrode	50×50 μm^2
ϵ_o	Silicon-oil permittivity	19×10^{-12} (F/m)
C_o	Capacitance of healthy microelectrodes	2.375×10^{-15} (F)
C_{d1}	Capacitance of partially degraded microelectrodes	2.380×10^{-15} (F)
C_{d2}	Capacitance of completely degraded microelectrodes	2.385×10^{-15} (F)

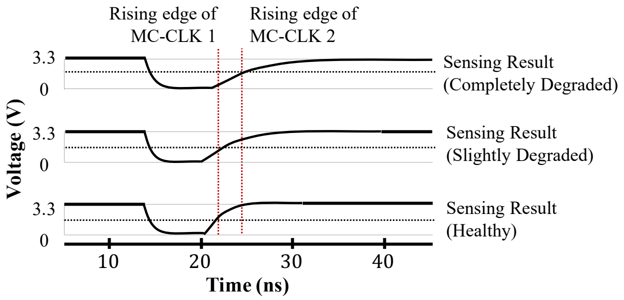


Fig. 2: Simulation results for the new microelectrode design.

Fig. 3 shows the simulation results for droplet sizes 3×3, 4×4, 5×5 and 6×6, and Manhattan distances $d \in \{1, 2, 3, 4, 5\}$. The general trend shows an inverse correlation relationship between the distance between two MCs and the number of actuations for each. The correlation coefficient is lowest for droplet size 3×3, and increases as the droplet size increases. For the same droplet size, however, the correlation coefficient is insensitive to the executed bioassay. The results suggest that, in general, two adjacent MCs are more likely to have similar number of actuations during execution. This can be explained by the fact that in MEDA biochips, MCs are typically actuated in clusters at any given actuation cycle. This implies that faulty MCs are more likely to appear in clusters as well. Hence, it is imperative to evaluate the proposed framework against clustered faults, as we cover later in Section VII.

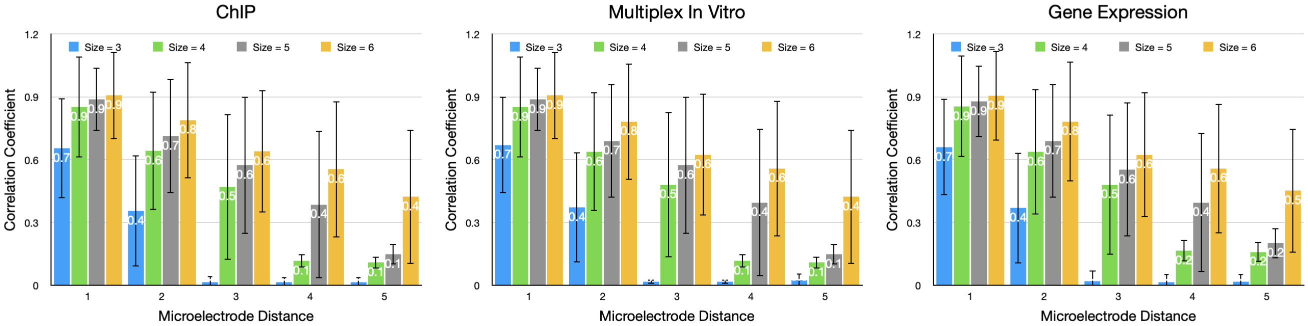


Fig. 3: Simulation results for the correlation coefficient versus the Manhattan distance between two microelectrodes.

IV. Microelectrode Degradation Model

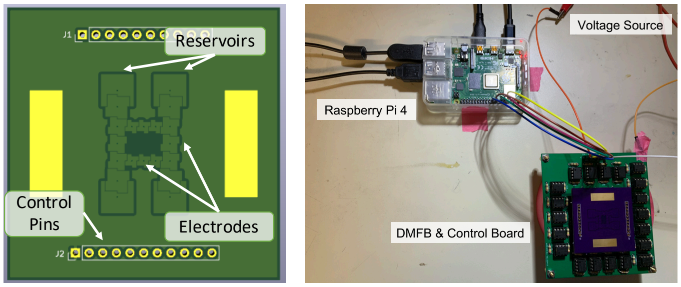
Given the new microelectrode hardware design, we can monitor the health level of individual MCs in realtime. In this section, we study the impact of the MC number of actuations and its degradation and health levels. Using the obtained results, we show how to utilize the MC health level to estimate the EWOD force exerted by the MC on an adjacent droplet.

A. Experimental Assessment of Electrode Degradation

Previous work has shown that charge trapping in a dielectric layer follows an exponential model [33], [28], [34], [35]. To independently validate this claim, we design an experiment where we monitor electrode degradation in a PCB-based digital microfluidic biochip (DMFB), which manipulates droplets using the same EWOD principle as MEDA biochips.

The DMFB contains electrodes in three sizes, v.i.z. $2{\times}2\,\mathrm{mm}^2$, $3{\times}3\,\mathrm{mm}^2$, and $4{\times}4\,\mathrm{mm}^2$; see Fig. 4(a). Four reservoir modules are placed on two sides of the biochip, and the modules can dispense different reagent droplets. Each electrode can be controlled individually, and these control signals come from a control board underneath the DMFB. For the fabricated DMFB, the activation/de-activation status of each electrode is controlled by a high voltage relay (Part No. Panasonic AQW212). Each high-voltage relay IC is controlled by a configuration bit, and these configuration bits are stored in the register ICs (Part No. Texas Instrument SN74AHC595). The hardware setup used to operate the digital microfluidic biochip is shown in Fig. 4(b). A micro-computer (Part No. Raspberry Pi 4) on the left is used to generate control signals to the control board. We used a voltage source of 1.5 KHz and 200 Vpp to actuate the electrodes. To avoid inducing high current, a resistor $R = 1$ MΩ is placed in series between each electrode and the high-voltage source.

We developed an actuation sequence for the electrodes that leads to repeated fluidic operations on the biochip. When we execute the actuation sequence on the DMFB, each electrode is actuated for 1 s for hundreds of times. After executing the actuation sequence, we actuated an electrode and measured the charging times needed using an oscilloscope. Because the electrode and the top plate form a capacitor, and a resistor is placed in series with the electrode, the charging path is a simple RC circuit. The effective capacitance of an electrode can be derived using the equation



(a) (b)

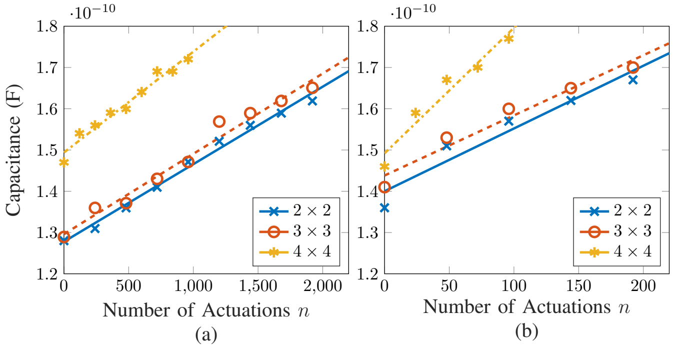
Fig. 4: (a) The fabricated DMFB and (b) the experimental setup.

$$V_C(t) = Vpp\left(1 - e^{-t/RC}\right),$$

where C is the effective capacitance of the electrode, V_C is the voltage of the electrode, and t is time. The degradation results are shown in Fig. 5(a). The results show that the capacitance of an electrode grows linearly as we repeatedly actuate the electrode. This increase in the capacitance can be attributed to charge trapping in the dielectric layer.

In addition to charge trapping in the dielectric layer, electrode degradation can also result from *residual charge*, which happens when an electrode is excessively actuated [36]. Excessive actuation of an electrode substantially increases the amount of charge that accumulates in the actuated electrode. If the next electrode is in turn actuated whereas the present electrode has residual charge, the droplet may not be moved toward the next direction as expected. We design an experiment that is similar to the previous experiment, but in which each electrode is actuated for 5 s instead of 1 s. The degradation results are shown in Fig. 5(b). The results show that the capacitance of an electrode also grows linearly as we repeatedly actuate the electrode, but the growth is much faster than that of the previous experiment.

Note that the microelectrode size of a MEDA biochip is different from the electrode size of a conventional DMFB. Nevertheless, our experimental results using fabricated devices with different electrode sizes show that the relative force on the electrodes follows the same degradation trend, irrespective of electrode size. As a MEDA biochip manipulates discrete fluids using the same EWOD principle as DMFBs, the electrodes and



(a)

(b)

Fig. 5: Experimental results for electrode degradation from (a) charge trapping and (b) residual charge.

relative force of a microelectrode on a MEDA biochip is also subject to the same degradation mechanism.

The work in [35] showed that the charge trapping phenomenon can be alleviated using a negative actuation voltage, i.e., AC actuation voltage. However, the mainstream commercialized DMFBs, such as Baebies, use DC actuation voltage (as in our work) because of simpler design, cheaper cost for the control circuit, and less concerns about Joule heating [37].

B. Microelectrode Health Model

Based on our experimental results for electrode degradation, in this section we develop a model that characterizes the relationship between the number of actuations n and the microelectrode health level. The EWOD force exerted by a microelectrode MC_{ij} (relative to the same EWOD force at full health) can be estimated as [16]

$$\bar{F}_{ij}^{(n)} \approx \left(V_{ij}^{(n)}/V_a\right)^2, \tag{1}$$

where $V_{ij}^{(n)}$ is the actuation voltage on MC_{ij} (potentially affected by the MC degradation), and V_a is the nominal actuation voltage. By plugging our experimental results to (1), the impact of the microelectrode number of actuations and the relative EWOD force is shown in Fig. 6. We note that there is a good fit between the model and the measured data. The model fitting results show that the relationship between the number of actuation n and the relative EWOD force $\bar{F}_{ij}^{(n)}$ can be modeled as

$$\bar{F}_{ij}^{(n)} \approx \tau^{2n/c}, \tag{2}$$

where $\tau \in [0,1]$ and $c \in \mathbb{R}$ are constants capturing the MC_{ij} degradation rate. For example, the parameter values in Fig. 6 are $(\tau_2, c_2) = (0.556, 822.7)$, $(\tau_3, c_3) = (0.543, 805.5)$, and $(\tau_4, c_4) = (0.530, 788.4)$, where $R^2_{\mathrm{adj}} > 0.94$ for all curves.

To deduce the relation between the measured health level $\mathbf{H}_{ij}$ and the number of actuations n, we define the *degradation* level of MC_{ij}, denoted by D_{ij}, as

$$D_{ij}^{(n)} = V_{ij}^{(n)}/V_a \approx \tau^{n/c} \in [0,1], \tag{3}$$

where $\mathbf{D}$ denotes the MC degradation matrix for the biochip. Moreover, given a number of bits $b \in \mathbb{N}_0$ for measuring the health level (recall that we use two bits in the MC design in Section III), we define MC_{ij} *health* level, H_{ij}, as

$$H_{ij}^{(n)} = \lfloor 2^b \cdot D_{ij}^{(n)} \rfloor = \lfloor 2^b \cdot \tau^{n/c} \rfloor,$$

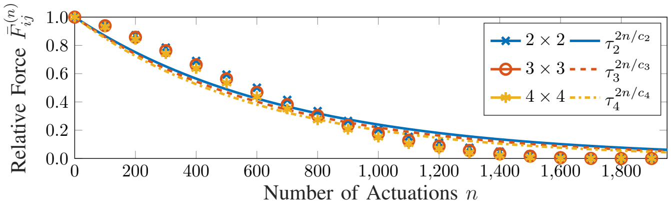


Fig. 6: Measured (markers) and fitted (lines) relative EWOD force $\bar{F}_{ij}$ versus the number of actuations n.

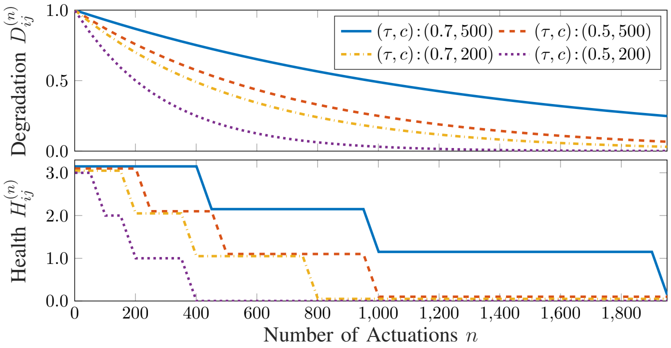


Fig. 7: Impact of the no. of actuations n on the actual degradation D_{ij} and observed MC health H_{ij} under different configurations.

where $\mathbf{H}$ denotes the MC health matrix for the biochip. Fig. 7 shows the impact of the number of actuations (n) on the observed $H_{ij}^{(n)}$ and the actual MC health $D_{ij}^{(n)}$ for various parameter configurations. The MC health exponentially decays as the number of actuations increase. The reliability model is valid for any general b, even though we use $b = 2$ for the results reported in this paper.

V. MEDA Biochip Model

A. Droplet Model

Consider a MEDA biochip with W × H MCs. A Boolean matrix of size W × H could be used to capture which MCs are covered by a droplet. However, a typical MEDA biochip can have over $1,800$ MCs [16], which would result in a model with the state-space that is too large for formal synthesis.

Hence, we use the properties of microfluidic operations to develop a scalable droplet actuation model. Specifically, we adopt an approach where we model a droplet using the underlying actuation pattern since the droplet size, shape and location are tightly coupled with the used actuation pattern. For example, actuation patterns typically take a rectangular, fully-filled form where free-roaming of droplets (i.e., leaving them without actuation) is not allowed; and under- or over-actuation of droplets is of no use. Thus, by restricting the state-space to the actuation patterns of interest, we significantly reduce the model size; thus, enabling runtime formal strategy synthesis.

We use $\mathbf{U}$ to indicate the biochip actuation matrix, where $U_{ij}^{(k)} \in \{0,1\}$ indicates whether MC_{ij} is actuated (1) or not actuated (0) at time k. We model a droplet as a tuple $\delta = (x_a, y_a, x_b, y_b) \in \mathbb{N}_0^4$, where (x_a, y_a) and (x_b, y_b) are the coordinates of the lower-left and upper-right corners (i.e., $x_b \geq x_a$, $y_b \geq y_a$), and $U_{ij} = 1$ for all $(i,j) \in [\![x_a, x_b]\!] \times [\![y_a, y_b]\!]$.

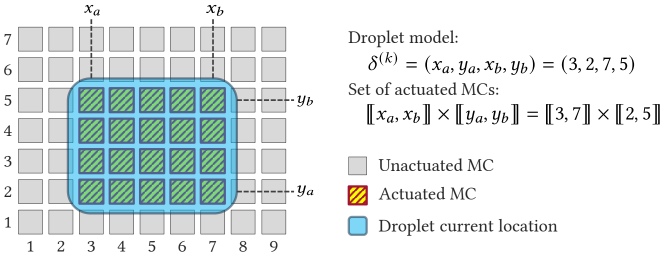

x_a
x_b
y_b
y_a
7
6
5
4
3
2
1
1 2 3 4 5 6 7 8 9
Droplet model:
$\delta^{(k)} = (x_a, y_a, x_b, y_b) = (3, 2, 7, 5)$
Set of actuated MCs:
$\llbracket x_a, x_b \rrbracket \times \llbracket y_a, y_b \rrbracket = \llbracket 3, 7 \rrbracket \times \llbracket 2, 5 \rrbracket$
Unactuated MC
Actuated MC
Droplet current location

Fig. 8: MEDA biochip segment with droplet δ at time k.

We use $\Delta \subset \mathbb{N}_0^4$ to denote the set of all possible droplets. We also use the center distance between two adjacent MCs as the unit length when describing droplet geometry. For a droplet $\delta = (x_a, y_a, x_b, y_b)$, the droplet width, height and area can be computed as $w = x_b - x_a + 1$, $h = y_b - y_a + 1$ and $A = wh$, respectively. We define a droplet's aspect ratio as $AR = w/h$.

Example 1 (Droplet Model)**.** *Fig. 8 shows a droplet* $\delta^{(k)} = (x_a, y_a, x_b, y_b)$ *where* $x_a = 3$, $y_a = 2$, $x_b = 7$ *and* $y_b = 5$, *i.e.,* $\delta^{(k)} = (3, 2, 7, 5)$. *The droplet's width, height, area and aspect ratio are* $w = 5$, $h = 4$, $A = 20$ *and* $AR = 5/4$, *respectively. The biochip actuation matrix satisfies that*

$$U_{ij}^{(k)} = \begin{cases} 1 & \forall (i,j) \in \llbracket 3, 7 \rrbracket \times \llbracket 2, 5 \rrbracket, \\ 0 & \textit{otherwise.} \end{cases}$$

B. Actuation Model

MEDA biochips support three classes of droplet manipulations: cardinal movement, ordinal movement, and shape morphing. We define the set of microfluidic actions as $\mathcal{A} = \mathcal{A}_d \cup \mathcal{A}_{dd} \cup \mathcal{A}_{dd'} \cup \mathcal{A}_{\downarrow} \cup \mathcal{A}_{\uparrow}$ where

- $\mathcal{A}_d = \{a_{\mathsf{N}}, a_{\mathsf{S}}, a_{\mathsf{E}}, a_{\mathsf{W}}\}$ are single-step movements in the cardinal directions,
- $\mathcal{A}_{dd} = \{a_{\mathsf{NN}}, a_{\mathsf{SS}}, a_{\mathsf{EE}}, a_{\mathsf{WW}}\}$ are double-step movements in the cardinal directions,
- $\mathcal{A}_{dd'} = \{a_{\mathsf{NE}}, a_{\mathsf{NW}}, a_{\mathsf{SE}}, a_{\mathsf{SW}}\}$ are movements in the ordinal directions,
- $\mathcal{A}_{\downarrow} = \{a_{\downarrow\mathsf{NE}}, a_{\downarrow\mathsf{NW}}, a_{\downarrow\mathsf{SE}}, a_{\downarrow\mathsf{SW}}\}$ are morphing transformations that increase droplet's width, and
- $\mathcal{A}_{\uparrow} = \{a_{\uparrow\mathsf{NE}}, a_{\uparrow\mathsf{NW}}, a_{\uparrow\mathsf{SE}}, a_{\uparrow\mathsf{SW}}\}$ are morphing transformations that increase droplet's height.

Single- and double-step movements aim to move the droplet a distance of a single and double units (i.e., one or two MCs in one cycle) in one of the cardinal directions, respectively. While $\mathcal{A}_d$, $\mathcal{A}_{dd}$, and $\mathcal{A}_{dd'}$ actions do not change droplet shape, $\mathcal{A}_{\downarrow}$ and $\mathcal{A}_{\uparrow}$ actions aim to change the droplet aspect ratio. Fig. 9 illustrates the microfluidic actions and their impact on droplets.

For an action $a \in \mathcal{A}$ and a droplet $\delta^{(k)} \in \Delta$, at time k, it holds that $\delta^{(k+1)} = a\left(\delta^{(k)}\right)$. We define the *frontier-set function* $Fr(\delta; a, d)$, capturing the subset of MCs that affect the movement of a droplet δ in direction $d \in \{\mathsf{N}, \mathsf{S}, \mathsf{E}, \mathsf{W}\}$ due to action a – i.e., $Fr(\bullet; a, d) : \Delta \to \mathcal{P}\left(\mathbb{N}_0^2\right)$. Table II shows the list of microfluidic actions and the respective frontier sets. Note that Fr is not defined for ordinal directions.

TABLE II: Microfluidic actions and their frontier sets, where $\delta = (x_a, y_a, x_b, y_b)$, $x^+ = x + 1$, $x^- = x - 1$, $d \in \{\mathsf{N}, \mathsf{S}\}$, and $d' \in \{\mathsf{E}, \mathsf{W}\}$.

Act.	$Fr(\delta; a, d)$	$\lvert Fr(\delta; a, d)\rvert$	$Fr(\delta; a, d')$	$\lvert Fr(\delta; a, d')\rvert$
a_{N}	$\llbracket x_a, x_b \rrbracket \times \llbracket y_b^+, y_b^+ \rrbracket$	$x_b - x_a + 1$	$\varnothing$	0
a_{S}	$\llbracket x_a, x_b \rrbracket \times \llbracket y_a^-, y_a^- \rrbracket$	$x_b - x_a + 1$	$\varnothing$	0
a_{E}	$\varnothing$	0	$\llbracket x_b^+, x_b^+ \rrbracket \times \llbracket y_a, y_b \rrbracket$	$y_b - y_a + 1$
a_{W}	$\varnothing$	0	$\llbracket x_a^-, x_a^- \rrbracket \times \llbracket y_a, y_b \rrbracket$	$y_b - y_a + 1$
a_{NE}	$\llbracket x_a^+, x_b^+ \rrbracket \times \llbracket y_b^+, y_b^+ \rrbracket$	$x_b - x_a + 1$	$\llbracket x_b^+, x_b^+ \rrbracket \times \llbracket y_a^+, y_b^+ \rrbracket$	$y_b - y_a + 1$
a_{NW}	$\llbracket x_a^-, x_b^- \rrbracket \times \llbracket y_b^+, y_b^+ \rrbracket$	$x_b - x_a + 1$	$\llbracket x_a^-, x_a^- \rrbracket \times \llbracket y_a^+, y_b^+ \rrbracket$	$y_b - y_a + 1$
a_{SE}	$\llbracket x_a^+, x_b^+ \rrbracket \times \llbracket y_a^-, y_a^- \rrbracket$	$x_b - x_a + 1$	$\llbracket x_b^+, x_b^+ \rrbracket \times \llbracket y_a^-, y_b^- \rrbracket$	$y_b - y_a + 1$
a_{SW}	$\llbracket x_a^-, x_b^- \rrbracket \times \llbracket y_a^-, y_a^- \rrbracket$	$x_b - x_a + 1$	$\llbracket x_a^-, x_a^- \rrbracket \times \llbracket y_a^-, y_b^- \rrbracket$	$y_b - y_a + 1$
$a_{\downarrow\mathsf{NE}}$	$\varnothing$	0	$\llbracket x_b^+, x_b^+ \rrbracket \times \llbracket y_a^+, y_b \rrbracket$	$y_b - y_a$
$a_{\downarrow\mathsf{NW}}$	$\varnothing$	0	$\llbracket x_a^-, x_a^- \rrbracket \times \llbracket y_a^+, y_b \rrbracket$	$y_b - y_a$
$a_{\downarrow\mathsf{SE}}$	$\varnothing$	0	$\llbracket x_b^+, x_b^+ \rrbracket \times \llbracket y_a, y_b^- \rrbracket$	$y_b - y_a$
$a_{\downarrow\mathsf{SW}}$	$\varnothing$	0	$\llbracket x_a^-, x_a^- \rrbracket \times \llbracket y_a, y_b^- \rrbracket$	$y_b - y_a$
$a_{\uparrow\mathsf{NE}}$	$\llbracket x_a^+, x_b \rrbracket \times \llbracket y_b^+, y_b^+ \rrbracket$	$x_b - x_a$	$\varnothing$	0
$a_{\uparrow\mathsf{NW}}$	$\llbracket x_a, x_b^- \rrbracket \times \llbracket y_b^+, y_b^+ \rrbracket$	$x_b - x_a$	$\varnothing$	0
$a_{\uparrow\mathsf{SE}}$	$\llbracket x_a^+, x_b \rrbracket \times \llbracket y_a^-, y_a^- \rrbracket$	$x_b - x_a$	$\varnothing$	0
$a_{\uparrow\mathsf{SW}}$	$\llbracket x_a, x_b^- \rrbracket \times \llbracket y_a^-, y_a^- \rrbracket$	$x_b - x_a$	$\varnothing$	0

Example 2 (Frontier Set)**.** *Fig. 10 shows a droplet* $\delta = (3, 2, 7, 5)$ *actuated under* a_{NE} *to initiate a movement in the* NE *direction. The set of MCs pulling the droplet to the east and north directions are* $Fr(\delta; a_{\mathsf{NE}}, \mathsf{E}) = \llbracket 8, 8 \rrbracket \times \llbracket 3, 6 \rrbracket$ *and* $Fr(\delta; a_{\mathsf{NE}}, \mathsf{N}) = \llbracket 4, 8 \rrbracket \times \llbracket 6, 6 \rrbracket$, *respectively.*

The degradation level of the MCs used in the movement (i.e., the MCs in the frontier set) impacts the EWOD driving force. Thus, a microfluidic action may not always result in the intended droplet movement. From (1), the relative EWOD force exerted on δ in direction $d \in \{\mathsf{N}, \mathsf{S}, \mathsf{E}, \mathsf{W}\}$ by action a can be estimated as

$$\bar{F}(\delta; a, d) = \sum_{(i,j) \in Fr(\delta; a, d)} \bar{F}_{ij} = \sum_{(i,j) \in Fr(\delta; a, d)} \tau^{2n_{ij}/c}.$$

As a larger EWOD force is more likely to move the droplet in the intended direction, the probability of whether an action a successfully moves droplet δ in direction d is a function of the degradation level of the MCs in $Fr(\delta; a, d)$. Let $\Sigma_d = \{\mathsf{N}, \mathsf{S}, \mathsf{E}, \mathsf{W}, \epsilon\}$ be the event space of executing $a_d \in \mathcal{A}_d$, where ϵ is the event of the droplet not moving. Assuming that all MCs in $Fr(\delta; a, d)$ equally contribute to the movement, the probability of an event E can be expressed as

$$p\left(E \mid \delta, a_d\right) = \begin{cases} \frac{\bar{F}(\delta; a_d, d)}{\lvert Fr(\delta; a_d, d)\rvert} & E = d, \\ 1 - \frac{\bar{F}(\delta; a_d, d)}{\lvert Fr(\delta; a_d, d)\rvert} & E = \epsilon, \\ 0 \quad \text{otherwise.} & \end{cases}$$

If a_d is successfully executed on $\delta^{(k)}$, the resulting droplet location is $\delta^{(k+1)} = a_d\left(\delta^{(k)}\right)$. Otherwise, the droplet location remains unchanged, i.e., $\delta^{(k+1)} = \delta^{(k)}$. In case of a double-step movement $a_{dd} \in \mathcal{A}_{dd}$, the probability that the second step is successful is conditioned on the success of the first step. Hence,

$$p\left(E \mid \delta, a_{dd}\right) = \begin{cases} \frac{\bar{F}(\delta; a_d, d)}{\lvert Fr(\delta; a_d, d)\rvert} \cdot \frac{\bar{F}(\delta'; a_d, d)}{\lvert Fr(\delta'; a_d, d)\rvert} & E = dd, \\ \frac{\bar{F}(\delta; a_d, d)}{\lvert Fr(\delta; a_d, d)\rvert} \cdot \left(1 - \frac{\bar{F}(\delta'; a_d, d)}{\lvert Fr(\delta'; a_d, d)\rvert}\right) & E = d, \\ 1 - \frac{\bar{F}(\delta; a_d, d)}{\lvert Fr(\delta; a_d, d)\rvert} & E = \epsilon, \\ 0 \quad \text{otherwise.} & \end{cases}$$

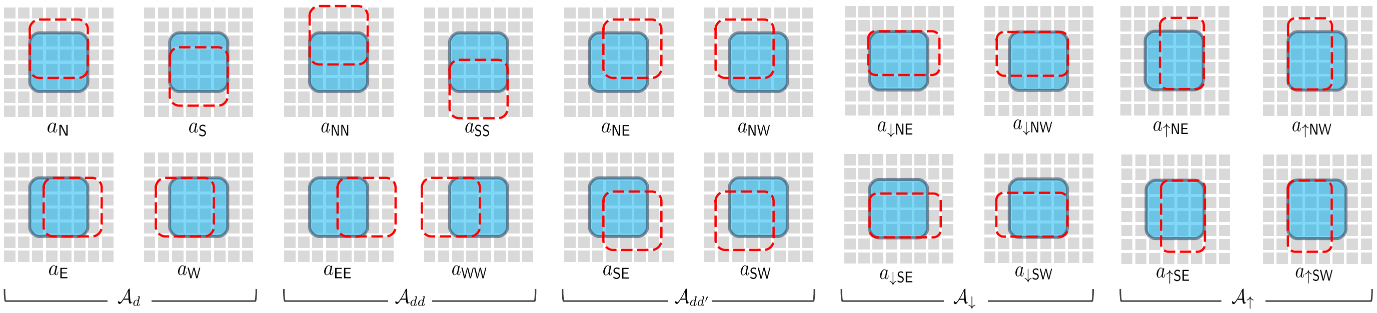


Fig. 9: Effect of microfluidic actions on droplets. Blue (solid) markers show initial locations. Red (dashed) markers show locations after successful execution.

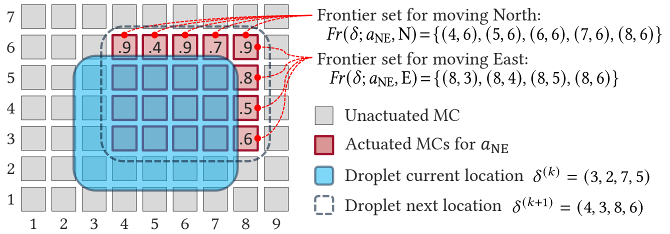

7
6
5
4
3
2
1
1
2
3
4
5
6
7
8
9
.9
.4
.9
.7
.9
.8
.5
.6
Frontier set for moving North:
$Fr(\delta; a_{\mathsf{NE}}, \mathrm{N}) = \{(4,6), (5,6), (6,6), (7,6), (8,6)\}$
Frontier set for moving East:
$Fr(\delta; a_{\mathsf{NE}}, \mathrm{E}) = \{(8,3), (8,4), (8,5), (8,6)\}$
Unactuated MC
Actuated MCs for a_{NE}
Droplet current location $\delta^{(k)} = (3,2,7,5)$
Droplet next location $\delta^{(k+1)} = (4,3,8,6)$

Fig. 10: MEDA biochip segment with droplet δ actuated under microfluidic action a_{NE} at time k.

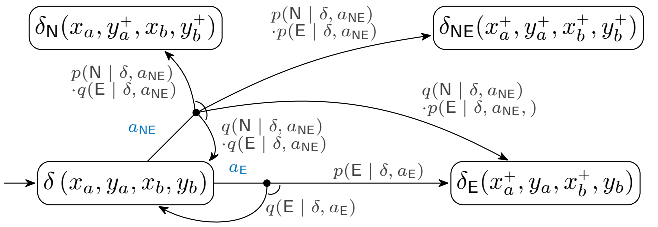

$\delta_{\mathsf{N}}(x_a, y_a^+, x_b, y_b^+)$
$p(\mathsf{N} \mid \delta, a_{\mathsf{NE}})$
$\cdot p(\mathsf{E} \mid \delta, a_{\mathsf{NE}})$
$\delta_{\mathsf{NE}}(x_a^+, y_a^+, x_b^+, y_b^+)$
$p(\mathsf{N} \mid \delta, a_{\mathsf{NE}})$
$\cdot q(\mathsf{E} \mid \delta, a_{\mathsf{NE}})$
$q(\mathsf{N} \mid \delta, a_{\mathsf{NE}})$
$\cdot p(\mathsf{E} \mid \delta, a_{\mathsf{NE}},)$
a_{NE}
$q(\mathsf{N} \mid \delta, a_{\mathsf{NE}})$
$\cdot q(\mathsf{E} \mid \delta, a_{\mathsf{NE}})$
a_{E}
$p(\mathsf{E} \mid \delta, a_{\mathsf{E}})$
$\delta\,(x_a, y_a, x_b, y_b)$
$\delta_{\mathsf{E}}(x_a^+, y_a, x_b^+, y_b)$
$q(\mathsf{E} \mid \delta, a_{\mathsf{E}})$

Fig. 11: Model for droplet $\delta = (x_a, y_a, x_b, y_b)$ showing two microfluidic actions a_{E} and a_{NE} and their transitions, where $q(E \mid \delta, a) = 1 - p(E \mid \delta, a)$.

where $\delta' = a_d(\delta)$ is the droplet location shifted by one step in the same direction. Similarly, the possible outcomes of an ordinal movement $a_{dd'} \in \mathcal{A}_{dd'}$ are moving in two directions, one direction, or none, captured by the event space $\Sigma_{dd'} = \{\mathsf{NE}, \mathsf{NW}, \mathsf{SE}, \mathsf{SW}\} \cup \Sigma_d$. The probability of each event can be expressed as

$$p(E \,|\, \delta, a_{dd'}) = \begin{cases} \frac{\bar{F}(\delta; a_{dd'}, d)}{|Fr(\delta; a_{dd'}, d)|} \cdot \frac{\bar{F}(\delta; a_{dd'}, d')}{|Fr(\delta; a_{dd'}, d')|} & E = dd', \\ \frac{\bar{F}(\delta; a_{dd'}, d)}{|Fr(\delta; a_{dd'}, d)|} \cdot \left(1 - \frac{\bar{F}(\delta; a_{dd'}, d')}{|Fr(\delta; a_{dd'}, d')|}\right) & E = d, \\ \left(1 - \frac{\bar{F}(\delta; a_{dd'}, d)}{|Fr(\delta; a_{dd'}, d)|}\right) \cdot \frac{\bar{F}(\delta; a_{dd'}, d')}{|Fr(\delta; a_{dd'}, d')|} & E = d', \\ \left(1 - \frac{\bar{F}(\delta; a_{dd'}, d)}{|Fr(\delta; a_{dd'}, d)|}\right) \cdot \left(1 - \frac{\bar{F}(\delta; a_{dd'}, d')}{|Fr(\delta; a_{dd'}, d')|}\right) & E = \epsilon, \\ 0 \qquad \text{otherwise.} \end{cases}$$

Fig. 11 illustrates part of the droplet model, where the actions a_{N} and a_{NE} and their probabilistic transitions are displayed. If a_{NE} is executed, the droplet can successfully move east to $\delta_{\mathsf{E}}(x_a^+, y_a, x_b^+, y_b)$ with probability $p(\mathsf{E} \mid \delta, a_{\mathsf{NE}})$, or remain at $\delta(x_a, y_a, x_b, y_b)$ with probability $q(\mathsf{E} \mid \delta, a_{\mathsf{NE}})$, where $q(E \mid \delta, a) = 1 - p(E \mid \delta, a)$.

Example 3 (Transition Probabilities)**.** *Continuing the running example, suppose that the frontier set MCs have degradation values* $D_{(8,3:6)} = (0.6, 0.5, 0.8, 0.9)$ *and* $D_{(4:8,6)} = (0.9, 0.4, 0.9, 0.7, 0.9)$ *as shown in Fig. 10. The probability of successfully moving the droplet in the* NE *direction is*

$$p\,(\mathsf{NE} \mid \delta, a_{\mathsf{NE}}) = \frac{.9{+}.4{+}.9{+}.7{+}.9}{5} \cdot \frac{.6{+}.5{+}.8{+}.9}{4} = 0.532.$$

Similarly, the probability of moving north is $p\,(\mathsf{N} \mid \delta, a_{\mathsf{NE}}) = 0.168$, *and east is* $p\,(\mathsf{E} \mid \delta, a_{\mathsf{NE}}) = 0.228$.

The morphing actions in $\mathcal{A}_{\downarrow}$ and $\mathcal{A}_{\uparrow}$ aim to decrease and increase the droplet's height, respectively. The difference amongst the actions of each set lies in the direction towards which the droplet's width and height are increased or decreased (see Fig. 9). For instance, $a_{\downarrow\mathsf{NE}}$ decreases the droplet's height by increasing its width towards the north-east direction. In contrast, $a_{\uparrow\mathsf{SW}}$ decreases the droplet's width by increasing its height towards the south-west direction. The probability of a morphing action being successful mainly depends on the set of MCs responsible for pulling the droplet. The frontier sets of morphing actions is listed in Table II.

In practice, the degree to which a droplet can be successfully morphed depends on its current size and shape. For instance, droplet aspect ratio may not go above $2/1$ or below $1/2$ to avoid unintentional splitting of the droplet. To model such constraints, we use *guards* on actions. A guard on action a is boolean expression g that represent a necessary condition for a to be enabled. Let $[r/1, 1/r]$, $r \geq 1$, be the allowed range for droplet aspect ratio. For shape morphing actions $a_{\uparrow} \in \mathcal{A}_{\uparrow}$ and $a_{\downarrow} \in \mathcal{A}_{\downarrow}$, we define two guards

$$g_{\uparrow}\colon \frac{y_b - y_a + 2}{x_b - x_a} \leq r \quad \text{and} \quad g_{\downarrow}\colon \frac{x_b - x_a + 2}{y_b - y_a} \leq r,$$

respectively. For example, for a maximum aspect ratio $r = 3/2$ and a droplet $\delta = (3, 2, 7, 5)$, $g_{\uparrow} = 1$, while $g_{\downarrow} = 0$. That is, $a_{\uparrow}$ is enabled while $a_{\downarrow}$ is disabled.

The model also supports moving two steps in a cardinal direction. Practically, a droplet can be reliably moved a distance no longer than half its length in one cycle. Hence, we impose a guard on double-step movements such that they are only enabled if such condition is satisfied. That is, double-step movements in the north or south directions are enabled for droplets with height $h \geq 4$, while double-step movements in the east or west directions are enabled for droplets with width $w \geq 4$. Consequently, the guards $g_{\mathsf{NN}}, g_{\mathsf{SS}}\colon y_b - y_a + 1 \geq 4$, and $g_{\mathsf{EE}}, g_{\mathsf{WW}}\colon x_b - x_a + 1 \geq 4$ are defined for a_{NN}, a_{SS}, a_{EE} and a_{WW}, respectively.

C. MEDA Biochip Model

While a droplet δ can be manipulated via various microfluidic actions as discussed earlier in Section V-B, the action outcomes are probabilistic. Moreover, such outcomes also depend on the MC health matrix $\mathbf{H}$. To accommodate the controller's choices and the probabilistic behaviors, we model the MEDA biochip using the stochastic-multiplayer games (SMGs) formalism.

Intuitively, in the MEDA SMG, denoted by $\mathcal{G}$, the game state is a triplet $s = (\delta, \mathbf{H}, \lambda)$, where $\lambda \in \{①, ②\}$ is the current player. Let $\mathcal{H} = \llbracket 0, 2^b - 1 \rrbracket^{\mathsf{W}\times\mathsf{H}}$ be the set of all possible $\mathbf{H}$. The state-space $S \subseteq \Delta \times \mathcal{H} \times \{①, ②\}$ captures all possible droplet locations, health states, and players' turn. A droplet controller constitutes the first player, ①, with an action set $\mathcal{A}_1 = \mathcal{A}$ (see Section V-B). The biochip degradation constitutes the second player, ②, with an action set $\mathcal{A}_2 = \mathcal{P}(\{a_{ij} | 1 \leq i \leq \mathsf{W}, 1 \leq j \leq \mathsf{H}\})$, where a_{ij} is the action of reducing H_{ij} by one. Note that ② can simultaneously take multiple actions (i.e., degrade multiple MCs at the same time). The initial state $s_0 = (\delta^{(0)}, \mathbf{H}^{(0)}, ①)$ defines the initial droplet location and health matrix. Finally, the MEDA SMG is formally defined as the tuple $\mathcal{G} = (S, \mathcal{A}_1 \cup \mathcal{A}_2, \gamma, s_0)$, where $\gamma \colon S \times \mathcal{A}_1 \cup \mathcal{A}_2 \times S \to [0, 1]$ is the transition probability function defined using the transition probabilities previously described in Section V-B.

Abstracting the biochip degradation as a player with nondeterministic actions serves two purposes. First, it allows for modeling a wide range of assumptions regarding the degradation behavior and fault-injection modes. Second, it enables the usage of two levels of model fidelity, one suitable for routing strategy synthesis, and the other for experimental simulations to validate the former.

Since the health matrix $\mathbf{H}$ is visible to the droplet controller, the resulting SMG is a full-information game. Hence, $\mathcal{G}$ can be used to synthesize droplet routing strategies as described in Section VI. For simulation, the same model is used, except that the health matrix $\mathbf{H}$ is substituted with the degradation matrix $\mathbf{D}$. The resulting SMG is an incomplete-information game since the droplet controller cannot observe $\mathbf{D}$. Further details on the simulation environment are covered in Section VII.

VI. Synthesis Framework

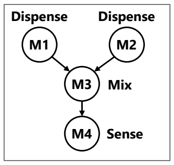
A typical bioassay is comprised of a series of droplet transportation and microfluidic operations that can be carried out on a MEDA biochip. In this section, we introduce a framework for adaptive strategy synthesis using the SMG-based MEDA model. We first show how to map various microfluidic operations into a set of droplet routing problems, called routing jobs. Next, we explore different ways to formalize requirements for routing strategies, and show how to use such requirements along with the SMG-based MEDA model to formally synthesize the routing strategies. Finally, we compare offline and online strategy synthesis methods.

A. Sequence Graphs and Microfluidic Operations

In this paper, we assume that a given bioassay is represented as a sequencing graph (SG), describing the list of microfluidic operations (MOs). We also assume that the SG is preprocessed by a planner that determines the dependencies and module placements of MOs, resulting in an MO list (e.g., see [16]). Each item in the list is described as $\mathsf{MO} = (\mathsf{type}, \mathsf{pre}, \mathsf{loc})$, where type is the MO type, pre is the list of predecessor MOs, and $\mathsf{loc} \in [1, \mathsf{W}] \times [1, \mathsf{H}]$ is the center location where the MO is executed. Table III summarizes the list of MO types and the associated number of input and output droplets.

TABLE III: List of microfluidic operations and the corresponding number of input and output droplets.

MO Type	Description	#Droplets (In, Out)
`dis`	Dispense a droplet (enter biochip)	$(0, 1)$
`out`/`dsc`	Output/Discard a droplet (exit biochip)	$(1, 0)$
`mix`	Mix two droplets into one	$(2, 1)$
`spt`	Split a droplet into two	$(1, 2)$
`dlt`	Dilute a droplet using another	$(2, 2)$
`mag`	Magnetically sense a droplet	$(1, 1)$



MO	type	pre	loc
M1	`dis`	$\varnothing$	$(17.5, 2.5)$
M2	`dis`	$\varnothing$	$(17.5, 28.5)$
M3	`mix`	{M1, M2}	$(10, 15)$
M4	`mag`	{M3}	$(40, 15)$

Fig. 12: Sequence graph example (left) and the corresponding MO list (right).

Example 4 (SG and MO List)**.** *Fig. 12 shows a simple sequence graph of four microfluidic operations.* M1 *and* M2 *are dispensing operations for* 4×4 *droplets* $(16, 1, 19, 4)$ *and* $(16, 27, 19, 30)$*, respectively, they have no predecessor operations. The center location for* M1*, for instance, is computed as* $((16 + 19)/2, (1 + 4)/2) = (17.5, 2.5)$. M3 *is a mixing operation between the two droplets from* M1 *and* M2*.*

B. Routing Jobs

To synthesize routing strategies, each MO is decomposed into a set of single-droplet routing problem, called a *routing job* (RJ). Each RJ stores the information necessary to synthesize a routing strategy for a single droplet. Formally, an RJ is a tuple $\mathsf{RJ} = (\delta_s, \delta_g, \delta_h)$, where δ_s is the droplet start location; δ_g is the droplet goal location; and δ_h is the hazard bounds for the routing job, defining the area within which the droplet is allowed to move. We design an *RJ helper* function that performs the aforementioned decomposition. Algorithm 1 summarizes the procedure that the RJ helper function follows to convert a given MO into a set of single-droplet routing jobs.

For dispensing operations, $\delta_s = (0, 0, 0, 0)$ since the initial location of the droplet is outside the biochip, while δ_g is where the droplet is dispensed. Note that the droplet size and shape are inferred from δ_g (see Section V-A). Since the dispensing operation is straightforward, the routing strategy is generated as a movement perpendicular to the edge from which the droplet is dispensed. For output and discard operations, however, routing is required for one droplet. The droplet's initial location is determined by the end location of the preceding MO, i.e., $\delta_s = \delta_{g_{\mathsf{pre}[0]}}$. The end location δ_g is the last on-chip location before the droplet exits the biochip through one of the four edges.

In a mixing operation, two droplets are routed from two different locations, δ_{s0} and δ_{s1}, to a single destination δ_g. In contrast, a splitting operation features two droplets that are routed from the same location δ_s to two different locations, δ_{g_0} and δ_{g_1}, that can be specified directly by the MO, or can be automatically placed at locations $\mathsf{loc}_{\mathsf{pre}[0]} + disp$ and $\mathsf{loc}_{\mathsf{pre}[0]} - disp$, respectively, where $disp$ is some displacement from the location where the splitting occurs. A dilution operation comprises mixing two droplets that start at δ_{s0} and δ_{s1}, followed by a splitting operation, resulting in δ_{g_0} and δ_{g_1}.

To compute the values of δ_s and δ_g, the droplet size needs to be known. For droplet-generating operations (i.e., dispensing), the desired size is already specified by the MO. For all other operations, the RJ helper computes the respective droplet sizes by first computing the droplet area and then obtain the droplet length and width that provide the minimum error in the computed area while satisfying the condition $|w - h| \leq 1$.

The hazard bounds δ_h represent the the rectangular area within which the routing can occur, while the droplet is forbidden from moving outside such area. Computing δ_h depends on the provided scheduler and resource allocation. In this work, we assume that the hazard bounds are computed as the rectangular area including both δ_s and δ_g, in addition to a safety margin of 3 MCs from each of the four sides to prevent accidental merging of droplets. That is, for $\delta_s = (x_a, y_a, x_b, y_b)$ and $\delta_g = (x'_a, y'_a, x'_b, y'_b)$, the hazard bounds are computed as $\delta_h = \textsc{Zone}(\delta_s, \delta_g)$ where

$$\begin{aligned}\textsc{Zone}(\delta_s, \delta_g) &:= (\min(x_a{-}3, x'_a{-}3, 1), \min(y_a{-}3, y'_a{-}3, 1),\\ &\quad \max(x_a{+}3, x'_a{+}3, \mathsf{W}), \max(y_a{+}3, y'_a{+}3, \mathsf{H})).\end{aligned}$$

More advanced computations of the hazard bounds can incorporate other information such as the droplet size and the number of concurrent operations.

Example 5 (RJ Helper)**.** *Continuing the previous example, let the dispensed droplets have size* 4×4. *Table IV shows the list of MOs and the associated RJs generated by the RJ helper for a MEDA biochip of size* 60×30. *Notice that* M3 *is decomposed into two routing jobs,* RJ3.0 *and* RJ3.1, *with the same goal location. The mixing operation results in a droplet area* $A = 32$, *which is approximated to a* 6×5 *actuation pattern. The location at which* M4 *occurs is centered at* $(40.5, 15.5)$, *and hence the target location for the corresponding routing job is estimated as* $(38, 14, 43, 18)$.

C. Routing Strategy Synthesis

Let $\mathcal{G} = (S, \mathcal{A}_1 \cup \mathcal{A}_2, \gamma, s_0)$ be the MEDA biochip model, $\mathsf{RJ} = (\delta_s, \delta_g, \delta_h)$ be a routing job, and $\mathbf{H}$ be the current health matrix. A droplet *routing strategy* is a mapping from ① states to the set of microfluidic actions $\mathcal{A}_1 = \mathcal{A}$ (see Section V-B), denoted by $\pi\colon S_1 \to A_1$. The *routing strategy problem* is concerned with finding a strategy π such that it satisfies a set of requirements. For the routing job RJ, the basic requirement is to for the droplet to eventually reach the goal location δ_g, i.e., a state $s = (\delta_g, \mathbf{H}, \lambda)$, while avoiding the hazard bounds δ_h.

To formalize this notion, we define two state labels, *goal* and *hazard*, that mark goal and hazard states, respectively.

Algorithm 1: MO-to-RJ Helper Procedure

```
Function MO_To_Rj(MO)
    Data: Microfluidic operation MO = (type, pre, loc)
    Result: Routing job list (RJ)
    switch type do
        case $dis$ do RJ[0] ← $(\mathbf{0}, \mathsf{loc}[0], \mathsf{loc}[0])$
        case $out, dsc, mag$ do RJ[0] ← $(\mathsf{pre}[0], \mathsf{loc}[0], \mathsf{loc}[0])$
        case $mix$ do
            RJ[0] ← $(\delta_{g_{\mathsf{pre}[0]}}, \mathsf{loc}[0], \textsc{Zone}(\mathsf{pre}[0], \mathsf{loc}[0]))$
            RJ[1] ← $(\delta_{g_{\mathsf{pre}[1]}}, \mathsf{loc}[0], \textsc{Zone}(\mathsf{pre}[1], \mathsf{loc}[0]))$
        case $spt$ do
            RJ[0] ← $(\delta_{g_{\mathsf{pre}[0]}}, \mathsf{loc}[0], \textsc{Zone}(\mathsf{pre}[0], \mathsf{loc}[0]))$
            RJ[1] ← $(\delta_{g_{\mathsf{pre}[0]}}, \mathsf{loc}[1], \textsc{Zone}(\mathsf{pre}[0], \mathsf{loc}[1]))$
        case $dlt$ do
            RJ[0] ← $(\delta_{g_{\mathsf{pre}[0]}}, \mathsf{loc}[0], \textsc{Zone}(\mathsf{pre}[0], \mathsf{loc}[0]))$
            RJ[1] ← $(\delta_{g_{\mathsf{pre}[1]}}, \mathsf{loc}[0], \textsc{Zone}(\mathsf{pre}[1], \mathsf{loc}[0]))$
            RJ[2] ← $(\delta_{g_{\mathsf{RJ}[0]}}, \mathsf{loc}[0], \textsc{Zone}(\mathsf{pre}[0], \mathsf{loc}[0]))$
            RJ[3] ← $(\delta_{g_{\mathsf{RJ}[1]}}, \mathsf{loc}[1], \textsc{Zone}(\mathsf{pre}[0], \mathsf{loc}[1]))$
    return (RJ)
```

Each label is a propositional formula over state-space variables that can be evaluated as true or false at a given state. For a routing job where $\delta_g = (x_{ag}, y_{ag}, x_{bg}, y_{bg})$ and $\delta_h = (x_{ah}, y_{ah}, x_{bh}, y_{bh})$, the two labels are defined as

$$\begin{aligned}goal\colon & (x_a \geq x_{ag}) \wedge (y_a \geq y_{ag}) \wedge (x_b \leq x_{bg}) \wedge (y_b \leq y_{bg}),\\ hazard\colon & (x_a < x_{ah}) \vee (y_a < y_{ah}) \vee (x_b > x_{bh}) \vee (y_b > y_{bh}).\end{aligned}$$

Note that *goal* utilizes inequalities rather than $\delta == \delta_g$ to allow for a less restrictive specification of the goal location. For instance, a 3×3 droplet may have a 5×5 goal location to indicate that the goal is to reach anywhere within the specified location. Using *goal* and *hazard*, the linear temporal logic (LTL) formula $\varphi\colon \Box(\neg hazard) \wedge \Diamond goal$ captures all possible executions that satisfy the routing requirement, where the temporal operator $\Diamond$ means *eventually true*, $\Box$ *always true*.[1]

Next, the LTL formula can be used to form a synthesis query. In this work, we explore the usage of two types of queries, namely, probabilistic and reward-based queries. The probabilistic query $\phi_p\colon \mathsf{P}_{\max=?}\,[\Box(\neg hazard) \wedge \Diamond goal]$ can be used to synthesize a strategy that maximizes the probability of satisfying φ. By feeding $\mathcal{G}$ and ϕ_p into a model checker, both an optimal strategy π_p (if existing) and the corresponding probability $p_{\max}$ are obtained.

Similarly, a reward-based query can be used for routing strategy synthesis. To this end, a reward function $r : S_1 \cup \mathcal{A}_1 \to \mathbb{R}_{\geq 0}$ is defined to reflect the reward associated with states and/or actions. For example, the reward function

$$r_k(a) = \begin{cases} 1 & \text{if } a \in \mathcal{A}_1,\\ 0 & \text{otherwise}\end{cases}$$

tracks the number of cycles (i.e., microfluidic actions) required to reach the goal location. The reward-based query

$$\phi_r\colon \mathsf{R}^{r_k}_{\min=?}\,[\Box(\neg hazard) \wedge \Diamond goal]$$

can be used to synthesize a strategy that minimized the total expected time (i.e., number of cycles) required to satisfy φ (i.e., to eventually reach the goal location without encountering a hazard zone). In contrast to ϕ_p, synthesizing using ϕ_r results in an optimal strategy π_r and the corresponding total expected

[1] More on LTL formulas can be found in [38].

TABLE IV: Example for converting MOs to RJs for a MEDA biochip of size $\mathsf{W} \times \mathsf{H} = 60 \times 30$.

MO	type	pre	loc	Size ($w \times h$)	Size Error	RJ	Start Location δ_s	Goal Location δ_g	Hazard Bounds δ_h
M1	`dis`	$\varnothing$	$(17.5, 2.5)$	$16\ (4 \times 4)$	0.0%	RJ1.0	$(00, 00, 00, 00)$	$(16, 01, 19, 04)$	$(13, 01, 22, 07)$
M2	`dis`	$\varnothing$	$(17.5, 28.5)$	$16\ (4 \times 4)$	0.0%	RJ2.0	$(00, 00, 00, 00)$	$(16, 27, 19, 30)$	$(13, 24, 22, 30)$
M3	`mix`	{M1, M2}	$(10.5, 15.5)$	$16\ (4 \times 4)$	0.0%	RJ3.0	$(16, 01, 19, 04)$	$(09, 14, 12, 17)$	$(06, 01, 22, 20)$
				$16\ (4 \times 4)$	0.0%	RJ3.1	$(16, 27, 19, 30)$	$(09, 14, 12, 17)$	$(06, 11, 22, 30)$
M4	`mag`	{M3}	$(40.5, 15.5)$	$32\ (6 \times 5)$	6.3%	RJ1.0	$(08, 14, 13, 18)$	$(38, 14, 43, 18)$	$(05, 11, 46, 21)$

Algorithm 2: Routing Strategy Synthesis Procedure

```
Function SYNTH(RJ, H)
    Data: RJ = $(\delta_s, \delta_g, \delta_h)$; health matrix $\mathbf{H}$
    Result: Strategy $\pi : \Delta \to \mathcal{A}$; expected completion time $k \in \mathbb{R}$
    Initialize model $\mathcal{G}$ using $\mathbf{H}$
    Let $goal\colon (x_a \geq x_{ag}) \wedge (y_a \geq y_{ag}) \wedge (x_b \leq x_{bg}) \wedge (y_b \leq y_{bg})$
    Let
      $hazard\colon (x_a < x_{ah}) \vee (y_a < y_{ah}) \vee (x_b > x_{bh}) \vee (y_b > y_{bh})$
    Let $\phi$: $\mathsf{R}_{\min=?}\,[\square\,(\neg hazard) \wedge \lozenge goal]$
    $(\pi, k) \leftarrow$ PRISMG $(\mathcal{G}, \phi, \delta_s)$
    return $(\pi, k)$
```

time $\mathbb{E}\,[r_k]$. However, the probability of reaching the target is not directly obtained.

Algorithm 2 describes the procedure to formally synthesize a routing strategy for a routing job RJ and health matrix $\mathbf{H}^{(k)}$. The synthesis starts by using both RJ and $\mathbf{H}^{(k)}$ to generate the associated SMG model $\mathcal{G}$, as described in Section V. Next, the procedure defines the labels for the goal states *goal* and hazard states *hazard* using the δ_g and δ_h, respectively. A synthesis query ϕ (such as ϕ_r) is formulated, and both $\mathcal{G}$ and ϕ are passed to a model checker (e.g., PRISM-games [39], represented as the function PRISMG) to synthesize an optimal routing strategy π and the expected number of cycles k. Note that if π does not exist, PRISMG returns $(\pi, k) = (\varnothing, \infty)$.

The computational complexity of the synthesizer is $\mathcal{O}\left((\mathsf{W} - w) \cdot (\mathsf{H} - h) \cdot |\mathcal{A}|\right)$, where $w \times h$ is the droplet size. To reduce the computational complexity, we apply partial order reduction to each routing job individually. First, for each routing job, the state-space is limited to the locations within the hazard bounds $\delta_h = (x_{ah}, y_{ah}, x_{bh}, y_{bh})$. We will use $\Delta_h \subseteq \Delta$ to denote the set of all possible droplet locations within δ_h. In the span of one routing job, the number of actuations required for a given MC is relatively small. Consequently, we can assume that the change in $\mathbf{H}$ is insignificant after a single action, rendering the order in which the degradation actions $\mathcal{A}_2$ occur irrelevant to the synthesis problem. By fixing $\mathbf{H}$ to the initial value $\mathbf{H}_0$, the SMG is reduced to the Markov decision process (MDP) $\mathcal{G}_{\mathsf{RJ}} = (\hat{S}, \mathcal{A}_1, \hat{\gamma}, \hat{s}_0)$ where $\hat{S} \subseteq \Delta_h \times \{\mathbf{H}_0\} \times \{①\}$, $\hat{\gamma}$ is defined by the rule

$$\hat{\gamma}\left((\delta, \mathbf{H}_0, ①), a, (\delta', \mathbf{H}_0, ①)\right) = \gamma\left((\delta, \mathbf{H}_0, ①), a, (\delta', \mathbf{H}_0, ②)\right)$$

and $\hat{s}_0 = (\delta_s, \mathbf{H}_0, ①)$. In this way, the computational complexity is reduced to $\mathcal{O}\left((w_h - x) \cdot (h_h - y) \cdot |\mathcal{A}_1|\right)$, where $w_h = x_{bh} - x_{ah} + 1$ and $h_h = y_{bh} - y_{ah} + 1$. The impact of the RJ area and droplet size on the runtime performance of the synthesis algorithm is further investigated later in Section VII-D.

D. Adaptive Routing Framework

The overall data-flow diagram of the proposed adaptive routing framework is shown in Fig. 13. The planner provides the list of MOs to the helper function to generate the list of routing jobs. Next, the synthesizer utilizes routing jobs RJ and the current health matrix $\mathbf{H}$ to synthesize the corresponding routing strategies. The low-level controller combines such strategies to generate the actuation pattern S at any time.

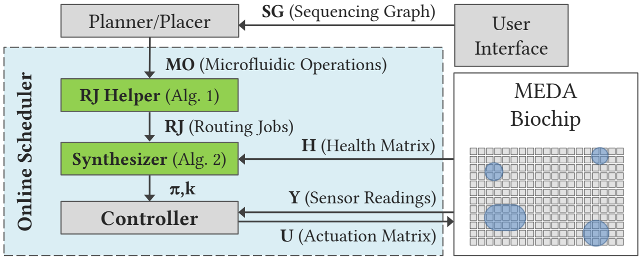


Fig. 13: Data flow diagram for the adaptive routing framework.

To synthesize a routing strategy, the synthesizer requires the health matrix value $\mathbf{H}$. Since the degradation characteristics of the MCs differ from one biochip to another, the exact value of $\mathbf{H}$ is unknown until the current routing job is due for execution. In an *online* scheduling scheme, strategies are synthesized on-demand and whenever $\mathbf{H}$ value is available. While such scheme is straightforward to implement, on-demand strategy synthesis introduces delays between subsequent microfluidic operations, which can be undesirable for time-sensitive bioassays. On the other hand, in an *offline* scheduling scheme, strategies are synthesized offline based on a range of expected values of $\mathbf{H}$. In runtime, the scheduler retrieves the pre-synthesized strategy that corresponds to the actual value of $\mathbf{H}$. While this scheme avoids synthesis delays, synthesizing, storing and retrieving strategies for all possible ranges of $\mathbf{H}$ is practically intractable. For instance, for a 20×20 biochip with $b = 2$ and a 4×4 droplet, the number of states is $|\hat{S}| > 10^{77}$.

To overcome this problem, we adopt a *hybrid* scheduling scheme that takes advantage of both the online and offline scheduling. In this scheme, a library of pre-synthesized strategies is first created offline for a range of droplet sizes and assuming no degradation. In runtime, the scheduler checks whether a strategy is available for the current $\mathbf{H}$ and retrieves it. Next, if a change in $\mathbf{H}$ value (i.e., degradation) is detected during the execution of a routing strategy, the scheduler relays the new value to the synthesizer to asynchronously resynthesize new strategies. Once the new strategies are available, they replace the previous ones since the value of a $\mathbf{H}$ element cannot regain its previous value once changed.

Algorithm 3 summarizes the procedure for the hybrid scheduler. First, the helper function populates the list of routing jobs for the given MO list, setting their statuses to `init`. In each cycle k, the scheduler reads the current sensor measurements

Algorithm 3: Hybrid Scheduler Procedure

```
Input: MO list $(\mathsf{MO})$; offline strategy library Lib
foreach $\mathsf{MO} \in (\mathsf{MO})$ do
    $(\mathsf{RJ})_{\mathsf{MO}} \leftarrow$ Mo_To_Rj$(\mathsf{MO})$, $\mathsf{state}_{\mathsf{MO}} \leftarrow \mathtt{init}$
$k \leftarrow 0$
while $\exists \mathsf{MO} \in (\mathsf{MO})$ s.t. $\mathsf{state}_{\mathsf{MO}} \neq \mathit{done}$ do
    $k$++, $\mathbf{U} \leftarrow \mathbf{0}$, Read $\mathbf{H}^{(k)}$
    Read $\mathbf{Y}^{(k)}$ and update the droplet location $\delta^{(k)}$ of each MO
    foreach $\mathsf{MO} \in (\mathsf{MO})$ do switch $\mathsf{state}_{\mathsf{MO}}$
        case $\mathit{init}$ do
            if $\forall \mathsf{MO}' \in (\mathsf{pre})_{\mathsf{MO}} : \mathsf{state}_{\mathsf{MO}'} == \mathit{done}$ then
                $\mathsf{state}_{\mathsf{MO}} \leftarrow \mathtt{active}$
                foreach $\mathsf{RJ} \in (\mathsf{RJ})_{\mathsf{MO}}$ : do
                    if a strategy for RJ exists in Lib then
                        $(\pi_{\mathsf{RJ}}, k_{\mathsf{RJ}}) \leftarrow$ Lib(RJ)
                    else
                        $(\pi_{\mathsf{RJ}}, k_{\mathsf{RJ}}) \leftarrow$ Synth$\big(\mathsf{RJ}, \mathbf{H}^{(k)}\big)$
                        Add $(\pi_{\mathsf{RJ}}, k_{\mathsf{RJ}})$ to Lib
        case $\mathit{active}$ do
            if $\forall \mathsf{RJ} \in (\mathsf{RJ})_{\mathsf{MO}} : \delta_g == \delta^{(k)}$ then
                $\mathsf{state}_{\mathsf{MO}} \leftarrow \mathtt{done}$
            else  $a \leftarrow \pi(\delta)$, $\mathbf{U}(a(\delta)) \leftarrow \mathbf{1}$
    Apply $\mathbf{U}$
```

$\mathbf{Y}^{(k)}$ and health matrix $\mathbf{H}^{(k)}$. Next, the scheduler checks if any MOs are ready for execution by confirming whether their corresponding predecessors are done and, if any, forwards the corresponding RJs to the synthesizer to retrieve their routing strategies. For an active MO, the optimal action a is retrieved from the current strategy (i.e., $a = \pi(\delta)$), and the corresponding MCs are set to be actuated (i.e., $\mathbf{U}(a(\delta)) = \mathbf{1}$). Finally, the actuation matrix $\mathbf{U}$ is applied to the biochip, and the process is repeated until all MOs finish execution.

VII. Experimental Evaluation

A. Experimental Setup

We implemented the online scheduler, helper, synthesizer, and a MEDA biochip simulator in MATLAB. The synthesizer automatically generated and passed routing jobs to PRISM-games [39] to obtain routing strategies. Results were obtained on an Intel Core i7 2.6 GHz CPU with 16 GB RAM.

The MEDA biochip simulator enables configuring the biochip width W and height H, in addition to the MC degradation behavior. In general, a microelectrode MC_{ij} is assigned uniformly sampled degradation parameters $c_{ij} \sim U(c_1, c_2)$ and $\tau_{ij} \sim U(\tau_1, \tau_2)$ (see Section IV-B). The simulator generates two types of MCs — normal and faulty — by sampling the degradation constants from two different uniform distributions, with the percentage of faulty MCs being configurable. The way in which faulty MCs are placed across the biochip depends on the fault-injection mode selected for the experiment. In *uniform* fault-injection mode, the faulty MCs are randomly placed. In *clustered* fault-injection mode, however, clusters of 2×2 faulty MCs are randomly placed.

We simulated six benchmark bioassays in our experiments: (i) Master-Mix, (ii) CEP, (iii) Serial Dilution [40], (iv) nucleosome immunoprecipitation (NuIP) [17], (v) COVID-RAT, and (vi) COVID-PCR. The CEP bioprotocol comprises three bioassays, namely, cell lysis, mRNA extraction, and mRNA purification. The NuIP bioprotocol is used for studying the epigenetic relationship between DNA and its supporting proteins [41]. Two COVID-19 tests, namely PCR-based and rapid antigen-based, are widely used to detect the presence of the SARS-CoV-2 virus or the body's response to infection [42]. PCR-based test, which is more accurate, detects small amounts of viral genetic material. Rapid antigen test, which is only effective for the first week of infection, detects the presence of viral proteins.

Fig. 14 shows the control flow of the simulation environment used in this section. First, the simulator instantiates a MEDA biochip according to the biochip configuration, the degradation parameters and the fault-injection mode selected for the experiment. Next, the online scheduler creates a new bioassay execution request based on the bioassay selected for the experiment, and further uses the helper to generate the list of routing jobs. Once the execution starts, at each cycle, the scheduler checks whether any MOs are ready for execution, reads the current health status from the biochip simulator. The synthesizer uses this information to generate the corresponding models and synthesis queries, passing them to the model checker to obtain the synthesized strategies. Subsequently, the controller uses the current biochip state and the synthesized strategies to populate the optimal action for each droplet, aggregate the actuation patterns for the current control cycle, and further pass the control matrix $\mathbf{U}$ to the biochip simulator. Using the current control matrix $\mathbf{U}$, both the number of actuations $\mathbf{N}$ and the actual degradation matrix $\mathbf{D}$ are updated accordingly. The next state of each droplet is randomly sampled from the probability distributions described in Section V-B, the droplets are checked for other conditions (e.g., merging and splitting), and the next cycle starts. The process continues until either the bioassay execution is successful, or the maximum number of cycles is reached, in which case the bioassay execution is aborted.

Two routing algorithms were implemented and used for the experiments in this section. The first (baseline) algorithm is unaware of degradation and generates the shortest-path strategy, minimizing the distance traveled by each droplet. The second (adaptive) algorithm follows the proposed synthesis framework to synthesize adaptive routing strategies based on the proposed framework (see Algorithm 2). Neither approach utilized the error-recovery techniques described in [43], [44] as our goal is to proactively avoid errors and the cost associated with error recovery.

B. Probability of Successful Completion

Since MEDA biochips are fabricated in a CMOS foundry, it is desirable to reuse them as much possible (e.g., for a panel of diagnostic tests for the same patient), as opposed to disposable devices fabricated on a plastic or glass substrate. Therefore, we examined the likelihood of successfully completing multiple runs of a bioassay for a given upper limit on the completion time ($k_{\max}$). We first simulated a fabricated MEDA biochip with 30×60 MCs [9]. Each MC followed the reliability model in (3), with degradation constants $c \sim U(200, 500)$ and $\tau \sim U(0.5, 0.9)$, randomly sampled to simulate microelectrode degradation. Once assigned, both c and τ remained constant during each set of experiments.

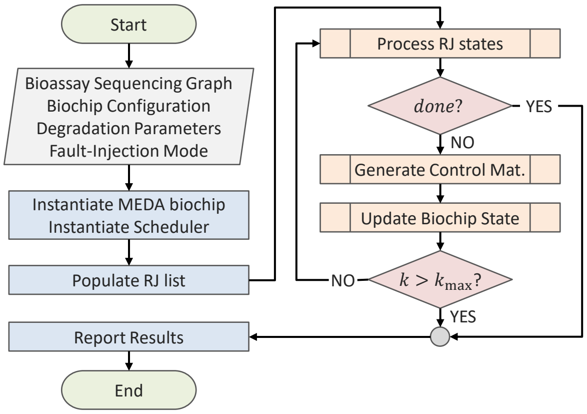


Fig. 14: Control flow for the simulation environment.

Fig. 15 shows that the proposed method ensures a significantly higher probability of successful bioassay completion (PoS) within the given limit on the time-to-result, especially for longer bioassays. For example, with $k_{\max} = 300$ cycles, the proposed approach guarantees the PoS for the Serial Dilution bioassay to be 0.8 compared to 0.1 for the baseline method. Even with more cycles (e.g., 320), the baseline method provides a PoS of only 0.7, while the PoS for the proposed method is 0.99. As expected, the proposed solution is more effective for longer bioassays. Lower $k_{\max}$ values imply fewer actuations per bioassay, increasing the number of successful executions before the biochip fully degrades.

C. Fault Injection During Bioassay Execution

In the next set of experiments, we randomly injected faults in MCs, wherein a droplet can get stuck at a group of faulty microelectrodes. The MCs were divided into two groups: normal and faulty. While both groups follow the degradation model described in Section V, a faulty MC exhibits a sudden failure at random actuation n, i.e., $D_{ij}^{(n)} = 0$. Moreover, two modes of fault injection were simulated: uniform and clustered. In the former, faulty MCs are randomly distributed across the biochip, while faults in the clustered mode appear as randomly-placed clusters of four adjacent MCs (2×2).

Fig. 16 compares the mean number of cycles (k) required to repeatedly execute each bioassay (referred to as a "trial") on the same MEDA biochip (i.e., the same degradation profile) under different routing strategies and fault-injection modes. A trial was terminated after five successful executions or if k exceeded the maximum allowed number of cycles $k_{\max} = 1{,}000$, in which case the execution was aborted because of excessive chip degradation. The probabilistic behavior in the actuation model implies that every trial uses potentially different droplet routes, therefore we also report standard deviation (SD) values.

The results show that the proposed adaptive method consistently requires fewer cycles to execute a bioassay compared to the baseline method. This gap becomes more pronounced when clustered faults are injected as such clusters act as roadblocks, obstructing droplet movements. In longer bioassays (e.g., Serial Dilution and NuIP), trials featuring the baseline method fail prematurely due to the excessive actuation of the same set of MCs. In contrast, the proposed method leads to successful bioassay execution by proactively avoiding degraded microelectrodes. The mean number of executions to first failure for the proposed method was greater than five in all bioassays, while the baseline method failed as early as in the first execution. Moreover, the relatively small variability (i.e., SD values) in k for the proposed method indicates its robustness against various distributions of fault occurrences.

D. Synthesis Runtime Performance

In this set of experiments, we examine the runtime performance of the synthesis framework. In particular, we observe the time required for adaptive strategy synthesis under various droplet and biochip sizes. Moreover, we study the impact of this time overhead on bioassay executions.

In the formal synthesis of strategies, the time required for synthesis is impacted by the state-space size, the number of transitions, and the synthesis query. Hence, we simulate a range of droplet sizes and hazard areas. Since all microfluidic operations are reduced to a number of routing jobs, it suffices to examine the routing jobs directly. The specific values of the health matrix does not impact the model size, except for cases where the health is 0 for a number of adjacent microelectrodes, which can lead to zero-probability transitions. Thus, we enforce the worst-case scenario by assuming a health matrix with no zero elements.

Table V shows the range of droplet and biochip sizes used in the experiments, and the corresponding model sizes. Note that the RJ area here refers to the size of the area specified by the hazard bounds of the given routing job, regardless of the total biochip size. As expected from the synthesis time complexity (see Section VI-C), for the same routing job area, the models of smaller droplets are larger in size. The time required to construct the model constitutes at least 90% of the total time required for strategy synthesis. For any droplet size, routing jobs with 20×20 or less area requires less than 3 seconds for strategy synthesis, which is a tolerable delay for most applications. On the other hand, RJs featuring larger areas may require as long as 10 seconds before the corresponding strategy is synthesized. Such delays can be catastrophic for time-sensitive bioassays; it can also lead to excessive degradation of MCs that are used to hold the droplets in place during that time.

TABLE V: Performance results for various droplet and biochip sizes.

Input Size (MC)		Model Size			Time (sec)		
RJ Area	Droplet	#States	#Transitions	#Choices	Construction	Synthesis	Total
10×10	3×3	67	1,913	697	0.602	0.074	0.675
10×10	4×4	52	1,419	525	0.590	0.060	0.650
10×10	5×5	39	997	377	0.566	0.058	0.624
10×10	6×6	28	647	253	0.548	0.051	0.600
20×20	3×3	327	10,813	3,737	2.429	0.175	2.604
20×20	4×4	292	9,599	3,325	2.233	0.159	2.392
20×20	5×5	259	8,457	2,937	2.105	0.139	2.244
20×20	6×6	228	7,387	2,573	1.959	0.136	2.095
30×30	3×3	787	26,720	9,066	9.599	0.282	9.881
30×30	4×4	732	24,793	8,418	9.193	0.271	9.464
30×30	5×5	679	22,938	7,794	8.594	0.264	8.858
30×30	6×6	628	21,155	7,194	8.445	0.253	8.698

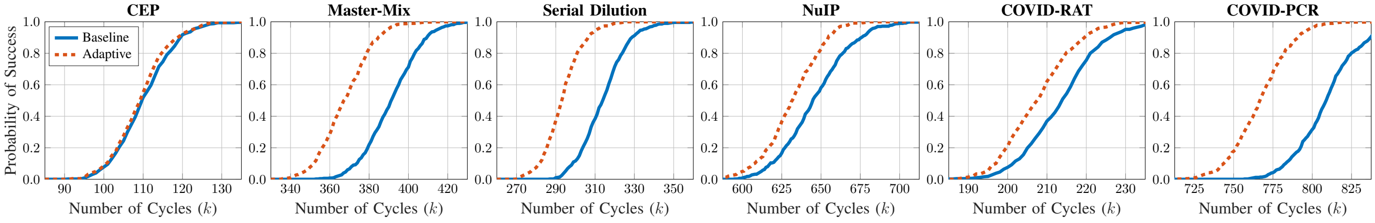


Fig. 15: Probability of successful bioassay completion versus the number of cycles allocated.

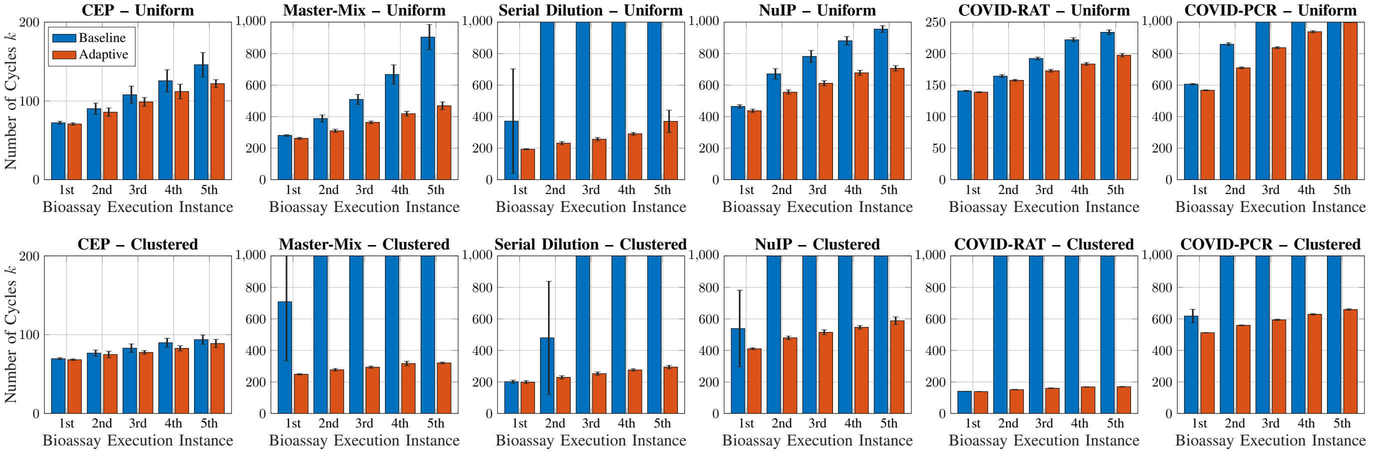


Fig. 16: Average number of cycles required to execute a bioassay under different routing strategies and fault-injection modes (standard deviations are indicated).

VIII. Conclusion

We have addressed the problem of microelectrode degradation in MEDA biochips by first introducing a new microelectrode-cell design that provides the health status of the microelectrodes. We have studied how the microelectrode degradation patterns are affected by droplet characteristics. We have also presented experimental results on electrode degradation for fabricated PCB prototypes, validating the degradation model throughout the paper. We have developed a stochastic game-based model for droplet manipulation that incorporates the health status, and used it to formally synthesize droplet routing strategies that dynamically adapt to the real-time microelectrode health information. Simulation results on four benchmark bioassays show that the proposed framework reduces the number of cycles required to successfully complete a bioassay in realistic microelectrode degradation scenarios. These results open the door for designing a scheduler that can optimize the order in which the microfluidic operations are executed in runtime.

References

[1] W.-L. Chou, P.-Y. Lee, C.-L. Yang, W.-Y. Huang, and Y.-S. Lin, "Recent advances in applications of droplet microfluidics," *Micromachines*, vol. 6, no. 9, pp. 1249–1271, 2015.

[2] R. S. Sista *et al.*, "Digital microfluidic platform to maximize diagnostic tests with low sample volumes from newborns and pediatric patients," *Diagnostics*, vol. 10, no. 1, p. 21, 2020.

[3] S. Huang, J. Connolly, A. Khlystov, and R. B. Fair, "Digital microfluidics for the detection of selected inorganic ions in aerosols," *Sensors*, vol. 20, no. 5, p. 1281, 2020.

[4] A. Ganguli, A. Mostafa, J. Berger, M. Y. Aydin, F. Sun, S. A. S. d. Ramirez, E. Valera, B. T. Cunningham, W. P. King, and R. Bashir, "Rapid isothermal amplification and portable detection system for SARS-CoV-2," *Proc. NAS*, 2020.

[5] C. Sheridan, "COVID-19 spurs wave of innovative diagnostics," *Nature biotechnology*, vol. 38, no. 7, pp. 769–772, 2020.

[6] NIH, "NIH delivering new COVID-19 testing technologies to meet U.S. demand," https://tinyurl.com/y4ynzhnq, [Online; accessed Aug. 2020].

[7] "Genmark Official Website," https://www.genmarkdx.com, 2020 [online].

[8] "Baebies Official Website," https://baebies.com, 2020 [online].

[9] K. Y.-T. Lai, Y.-T. Yang, and C.-Y. Lee, "An intelligent digital microfluidic processor for biomedical detection," *Journal of Signal Processing Systems*, vol. 78, no. 1, pp. 85–93, 2015.

[10] C. Quilliet and B. Berge, "Electrowetting: a recent outbreak," *Current Opinion in Colloid & Interface Science*, vol. 6, no. 1, pp. 34–39, 2001.

[11] Y. Ho, G. Wang, K. Y.-T. Lai, Y.-W. Lu, K.-M. Liu, Y.-M. Wang, and C.-Y. Lee, "Design of a micro-electrode cell for programmable lab-on-cmos platform," in *Proc. ISCAS*, 2016, pp. 2871–2874.

[12] T. Xu and K. Chakrabarty, "Fault modeling and functional test methods for digital microfluidic biochips," *IEEE TBioCAS*, vol. 3, pp. 241–253, 2009.

[13] F. Su, S. Ozev, and K. Chakrabarty, "Ensuring the operational health of droplet-based microelectrofluidic biosensor systems," *IEEE Sensors Journal*, vol. 5, no. 4, pp. 763–773, 2005.

[14] H. Verheijen and M. Prins, "Reversible electrowetting and trapping of charge: model and experiments," *Langmuir*, vol. 15, pp. 6616–6620, 1999.

[15] A. I. Drygiannakis, A. G. Papathanasiou, and A. G. Boudouvis, "On the connection between dielectric breakdown strength, trapping of charge, and contact angle saturation in electrowetting," *Langmuir*, vol. 25, pp. 147–152, 2009.

[16] Z. Zhong, Z. Li, K. Chakrabarty, T.-Y. Ho, and C.-Y. Lee, "Micro-electrode-dot-array digital microfluidic biochips: Technology, design automation, and test techniques," *IEEE TBioCAS*, vol. 13, no. 2, pp. 292–313, 2018.

[17] T.-C. Liang, Z. Zhong, Y. Bigdeli, T.-Y. Ho, K. Chakrabarty, and R. Fair, "Adaptive droplet routing in digital microfluidic biochips using deep reinforcement learning," in *Proc. ICML*, 2020.

[18] O. Keszocze, R. Wille, T.-Y. Ho, and R. Drechsler, "Exact one-pass synthesis of digital microfluidic biochips," in *Proc. DAC*, 2014, pp. 1–6.

[19] K. Chakrabarty, R. B. Fair, and J. Zeng, "Design tools for digital microfluidic biochips: toward functional diversification and more than moore," *IEEE TCAD*, vol. 29, no. 7, pp. 1001–1017, 2010.

[20] K. O'Neal, D. Grissom, and P. Brisk, "Resource-constrained scheduling for digital microfluidic biochips," *ACM JETC*, vol. 14, pp. 1–26, 2017.

[21] T. Xu and K. Chakrabarty, "Integrated droplet routing in the synthesis of microfluidic biochips," in *Proc. DAC*, 2007, pp. 948–953.

[22] O. Keszocze, R. Wille, K. Chakrabarty, and R. Drechsler, "A general and exact routing methodology for digital microfluidic biochips," in *Proc. ICCAD*, 2015, pp. 874–881.

[23] Q. Wang, Z. Li, H. Cheong, O.-S. Kwon, H. Yao, T.-Y. Ho, K. Shin, B. Li, U. Schlichtmann, and Y. Cai, "Control-fluidic codesign for paper-based digital microfluidic biochips," in *Proc. ICCAD*, 2016, pp. 1–8.

[24] J. McDaniel, Z. Zimmerman, D. Grissom, and P. Brisk, "PCB escape routing and layer minimization for digital microfluidic biochips," *IEEE TCAD*, vol. 36, no. 1, pp. 69–82, 2016.

[25] O. Keszocze, Z. Li, A. Grimmer, R. Wille, K. Chakrabarty, and R. Drechsler, "Exact routing for micro-electrode-dot-array digital microfluidic biochips," in *Proc. ASP-DAC*, 2017, pp. 708–713.

[26] Y. Luo, K. Chakrabarty, and T.-Y. Ho, "Error recovery in cyberphysical digital microfluidic biochips," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 1, pp. 59–72, 2012.

[27] ——, "Real-time error recovery in cyberphysical digital-microfluidic biochips using a compact dictionary," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 12, pp. 1839–1852, 2013.

[28] S. Zafar, A. Callegari, E. Gusev, and M. V. Fischetti, "Charge trapping related threshold voltage instabilities in high permittivity gate dielectric stacks," *Journal of Applied Physics*, vol. 93, no. 11, pp. 9298–9303, 2003.

[29] C.-H. Lee, S.-H. Hur, Y.-C. Shin, J.-H. Choi, D.-G. Park, and K. Kim, "Charge-trapping device structure of SiO_2/SiN/high-k dielectric Al_2O_3 for high-density flash memory," *Applied Physics Letters*, vol. 86, no. 15, p. 152908, 2005.

[30] C. Besset, S. Bruyere, S. Blonkowski, S. Cremer, and E. Vincent, "MIM capacitance variation under electrical stress," *Microelectronics Reliability*, vol. 43, no. 8, pp. 1237–1240, 2003.

[31] B. Razavi, K. F. Lee, and R.-H. Yan, "A 13.4-GHz CMOS frequency divider," in *Proc. ISSCC*, 1994, pp. 176–177.

[32] T.-C. Liang, Z. Zhong, M. Pajic, and K. Chakrabarty, "Extending the lifetime of meda biochips by selective sensing on microelectrodes," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 11, pp. 3531–3543, 2020.

[33] W. K. Meyer and D. L. Crook, "Model for oxide wearout due to charge trapping," in *Proc. IRPS*. IEEE, 1983, pp. 242–247.

[34] Y. Liu, A. Shanware, L. Colombo, and R. Dutton, "Modeling of charge trapping induced threshold-voltage instability in high-κ gate dielectric FETs," *IEEE Electron Device Letters*, vol. 27, no. 6, pp. 489–491, 2006.

[35] C. Dong, T. Chen, J. Gao, Y. Jia, P.-I. Mak, M.-I. Vai, and R. P. Martins, "On the droplet velocity and electrode lifetime of digital microfluidics: voltage actuation techniques and comparison," *Microfluidics and Nanofluidics*, vol. 18, pp. 673–683, 2015.

[36] T.-W. Huang, T.-Y. Ho, and K. Chakrabarty, "Reliability-oriented broadcast electrode-addressing for pin-constrained digital microfluidic biochips," in *Proc. ICCAD*. IEEE, 2011, pp. 448–455.

[37] V. Srinivasan, V. K. Pamula, and R. B. Fair, "An integrated digital microfluidic lab-on-a-chip for clinical diagnostics on human physiological fluids," *Lab on a Chip*, vol. 4, no. 4, pp. 310–315, 2004.

[38] C. Baier, J.-P. Katoen, and K. G. Larsen, *Principles of Model Checking*. MIT press, 2008.

[39] M. Kwiatkowska, G. Norman, D. Parker, and G. Santos, "Prism-games 3.0: Stochastic game verification with concurrency, equilibria and time," in *Proc. CAV*, 2020, pp. 475–487.

[40] M. Elfar, Z. Zhong, Z. Li, K. Chakrabarty, and M. Pajic, "Synthesis of error-recovery protocols for micro-electrode-dot-array digital microfluidic biochips," *ACM TECS*, vol. 16, pp. 1–22, 2017.

[41] P. Collas, "The current state of chromatin immunoprecipitation," *Molecular Biotechnology*, vol. 45, no. 1, pp. 87–100, 2010.

[42] G. Guglielmi, "Fast coronavirus tests: what they can and can't do." *Nature*, vol. 585, no. 7826, pp. 496–498, 2020.

[43] Z. Li, K. Y.-T. Lai, J. McCrone, P.-H. Yu, K. Chakrabarty, M. Pajic, T.-Y. Ho, and C.-Y. Lee, "Efficient and adaptive error recovery in a micro-electrode-dot-array digital microfluidic biochip," *IEEE TCAD*, vol. 37, pp. 601–614, 2017.

[44] Z. Zhong, Z. Li, and K. Chakrabarty, "Adaptive and roll-forward error recovery in MEDA biochips based on droplet-aliquot operations and predictive analysis," *IEEE TMSCS*, vol. 4, pp. 577–592, 2018.

Mahmoud Elfar received the B.Sc. degree in mechatronics from Ain Shams University, Cairo, Egypt. He is currently working toward the Ph.D. degree in computer engineering at Duke University, Durham, NC, USA. Previously, he was an R&D Engineer at Schneider Electric, and a Software Engineer at Valeo. He is currently a Research Assistant with the Duke Cyber-Physical Systems Lab. His research interests include formal methods, model checking techniques and human-robot interaction, and their applications in cyber-physical systems.

Tung-Che Liang received his B.S. degree in Electronics Engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2014, and the M.S. and Ph.D. degrees in Electrical and Computer Engineering from Duke University, Durham, NC, USA, in 2020 and 2021, respectively. He is currently a DFT engineer with Nvidia Inc. He was with Synopsys Inc. as an R&D engineer, a yield & diagnosis intern at Intel, and a DFT intern at NVIDIA Inc. and Apple Inc. His research interests include deep learning, design automation, and microfluidic system security.

Krishnendu Chakrabarty received the B. Tech. degree from the Indian Institute of Technology, Kharagpur, in 1990, and the Ph.D. degree from the University of Michigan, Ann Arbor, in 1995. He is now the John Cocke Distinguished Professor and Department Chair of Electrical and Computer Engineering at Duke University, Durham, NC, USA.

Prof. Chakrabarty is a recipient of the NSF CAREER award, the ONR Young Investigator award, the Humboldt Research Award from the Alexander von Humboldt Foundation, Germany, the IEEE Transactions on CAD Donald O. Pederson Best Paper Award (2015), the IEEE Transactions on VLSI Systems Prize Paper Award (2021), and the ACM Transactions on Design Automation of Electronic Systems Best Paper Award (2017). He is also a recipient of the IEEE Computer Society Technical Achievement Award (2015), the IEEE Circuits and Systems Society Charles A. Desoer Technical Achievement Award (2017), the IEEE Circuits and Systems Society Vitold Belevitch Award (2021), and the Semiconductor Research Corporation Technical Excellence Award (2018). He was an Invitational Fellow of the Japan Society for the Promotion of Science in the "Short Term" category ("at the Nobel Prize level") in 2018. His current research projects include: design-for-testability of integrated circuits; microfluidic biochips; hardware security; neuromorphic computing systems; AI for healthcare. He is a Fellow of IEEE, ACM, and AAAS.

Miroslav Pajic received the Dipl. Ing. and M.S. degrees in electrical engineering from the University of Belgrade, Serbia, in 2003 and 2007, respectively, and the M.S. and Ph.D. degrees in electrical engineering from the University of Pennsylvania, Philadelphia, PA, USA, in 2010 and 2012, respectively.

He is currently the Dickinson Family Associate Professor in the Department of Electrical and Computer Engineering at Duke University. His research interests focus on the design and analysis of high-assurance cyber-physical systems with varying levels of autonomy and human interaction, at the intersection of (traditional) areas of embedded systems, AI, learning and controls, formal methods, and robotics.

Dr. Pajic received various awards including the ACM SIGBED Early-Career Award, IEEE TCCPS Early-Career Award, NSF CAREER Award, ONR Young Investigator Program Award, ACM SIGBED Frank Anger Memorial Award, Joseph and Rosaline Wolf Best Dissertation Award from Penn Engineering, IBM Faculty Award, as well as seven Best Paper and Runner-up Awards at major cyber-physical systems venues.