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

Mahmoud Elfar, Tung-Che Liang, Krishnendu Chakrabarty, and Miroslav Pajic
Duke University, Durham NC, USA
{mahmoud.elfar, tung.che.liang, krishnendu.chakrabarty, miroslav.pajic}@duke.edu

***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].

Micro-electrode-dot-array (MEDA) biochips have been proposed in recent years to further advance DMFB technology [7]. A MEDA biochip manipulates fluids as discrete droplets of picoliter volume using the principle of electro-wetting-on-dielectric (EWOD) on a two-dimensional array of microelectrodes [8]. 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 [9].

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 [10]. Some of these are related to manufacturing defects; post-fabrication testing can be used to screen for such defects [11]. However, charge trapping in the dielectric layer and degradation of the insulator can result in microelectrode degradation [12], [13]. If an electrode is degraded during bioassay execution, fluidic operations associated with this electrode will fail, resulting in bioassay failure [14], [15].

Hence, to reliably execute bioassays on MEDA biochips, we must proactively avoid the use of degraded microelectrodes. In this paper, we first present a new hardware design that can provide dynamic health information about microelectrodes. We next present a scalable formal synthesis method for droplet routing that can dynamically change droplet transportation routes, based on the real-time health information obtained from microelectrodes. The specific contributions of this paper are the following:

- We introduce a MEDA biochip hardware design that allows fine-grained sensing of individual microelectrodes.
- We develop a scalable formal stochastic game-based model for droplet manipulation in a MEDA biochip. This model utilizes the fine-grained sensing results from the new hardware design.
- We introduce a formal synthesis framework where the the above model is utilized to automatically synthesize droplet routing strategies that maximize the likelihood of successful bioassay execution.
- We develop a MEDA biochip simulator to evaluate the proposed framework and present simulation results for four benchmark bioassays.

II. Background and Microelectrode Cell Design

A. Bioprotocol Synthesis on DMFBs

In the DMFB synthesis flow [16], 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 [17], [18]. Droplets need to be transported as part of the bioprotocol synthesized on the DMFB. Various droplet-routing methods have been proposed in the literature [19], [20], [21], [22], including techniques

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

978-3-9819263-5-4/DATE21/©2021 EDAA

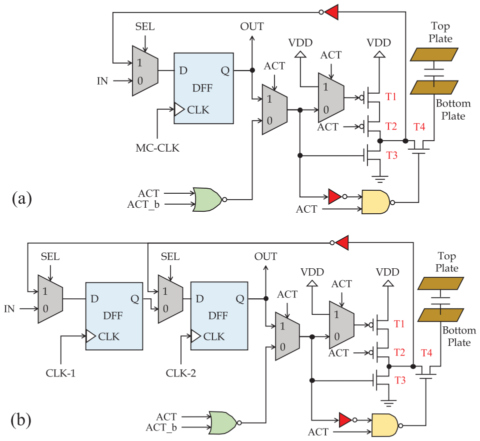

Fig. 1: Schematics of (a) the original and (b) the new proposed MC in MEDA.

TABLE I: Notation used for the simulation.

Symbols	Description	Values
A	Area of a microelectrode	$50\times50\,\mu m^2$
ϵ_o	Silicon-oil permittivity	19×10^{-12} (F/m)
C_o	Capacitance of a healthy microelectrode	2.375×10^{-15} (F)
C_{d1}	Capacitance of a partially degraded microelectrode	2.380×10^{-15} (F)
C_{d2}	Capacitance of a completely degraded microelectrode	2.385×10^{-15} (F)

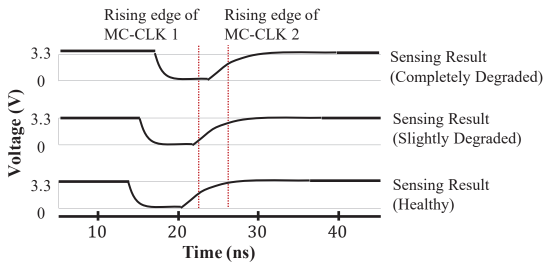

Fig. 2: Simulation results for the new MC.

that are specific to MEDA [23], [14]. However, these methods do not consider electrode degradation. Recently, reinforcement learning has been proposed to transport droplets in a reliable manner [15]. However, this approach does not monitor the dynamic health condition of electrodes and therefore fluidic operations associated with degraded electrodes may still fail.

B. 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 [14]. 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*.

C. 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 [13], [24], [25]. 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 plate [26]; 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 newly added DFF is able to capture this difference by registering a different value compared to the original 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 III–IV.

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 [27]. 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 ($\sim 27\,\mu m^2$) is much less than the area of a microelectrode ($2{,}500\,\mu m^2$) minus the area of the electronics underneath it ($\sim 88.2\,\mu m^2$) [7]; the microfluidics part clearly dominates the overall area of the MC.

III. MEDA Biochip Model

Notation: $\mathbb{N}_0$ is the set of non-negative integers. For $a, b \in \mathbb{N}_0$, $[\![a, b]\!]$ is the discrete interval $\{x \mid x \in \mathbb{N}_0, a \leq x \leq b\}$, and $[\![(a_1, a_2), (b_1, b_2)]\!]$ denotes the 2-dimensional discrete interval $\{(x, y) \mid (x, y) \in [\![a_1, b_1]\!] \times [\![a_2, b_2]\!]\}$. For a variable x, $x^{(k)}$ denotes its value at time $k \in \mathbb{N}_0$. We use bold symbols for matrices, e.g., $\mathbf{M} = (M_{ij}) \in \mathbb{R}^{m \times n}$; M_{ij} is the element in the ith row and jth column. For a set A, $\mathcal{P}(A)$ is its power set.

A. Droplet Model

Consider a MEDA biochip with $\mathsf{W} \times \mathsf{H}$ MCs. A Boolean matrix of size $\mathsf{W} \times \mathsf{H}$ could be used to capture which MCs are covered by a droplet. However, a typical MEDA biochip can have over 1,800 MCs [14], resulting in a model with a 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 model a droplet using the underlying actuation pattern as the droplet size, shape and location are tightly coupled with such 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. By restricting the state-space to the actuation patterns of interest, we significantly reduce the model size; thus, enabling runtime formal strategy synthesis.

Formally, 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. Furthermore, we model a droplet as $\delta = (x_a, y_a, x_b, y_b) \in \mathbb{N}_0^4$, 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 $(i, j) \in [\![(x_a, y_a), (x_b, y_b)]\!]$. We use $\Delta \subset \mathbb{N}_0^4$ to denote the set of all possible droplets.

B. MC Reliability Model

Previous work has shown that charge trapping in a dielectric layer follows an exponential model, and such models have been experimentally verified [28], [24], [29]. To be consistent with the experimental results reported for DMFBs in [30], we model the reliability of an MC as

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

where n is the number of times MC_{ij} was actuated, and $\tau \in [0, 1]$ and $c \in \mathbb{R}$ are constants capturing the MC_{ij} degradation rate. From (1), the (degraded) actuation voltage on MC_{ij} can be denoted as $V_{ij}^{(n)} = V_a D_{ij}^{(n)} = V_a \tau^{n/c}$, where V_a is the nominal actuation voltage. The relative EWOD force exerted by MC_{ij} on an adjacent droplet can be estimated [14]:

$$\bar{F}_{ij}^{(n)} \approx \left(V_{ij}^{(n)} / V_a\right)^2 = \tau^{2n/c}. \tag{2}$$

Finally, the health level of MC_{ij} is observable via variable $H_{ij}^{(n)} = \lfloor 2^b \cdot D_{ij}^{(n)} \rfloor = \lfloor 2^b \cdot \tau^{n/c} \rfloor$, where $b \in \mathbb{N}_0$ is the number of bits used to measure the health level (recall that we use two bits in the MC design of Sec. II), and $\mathbf{H} = (H_{ij})$ is the MC health matrix for the biochip. Fig. 3 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 reliability model is valid for any general b, even though we use $b = 2$ for the simulations results reported in this paper.

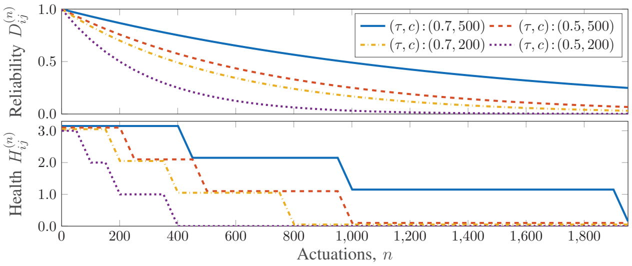

Fig. 3: No. of actuations n of MC_{ij} versus its reliability and health.

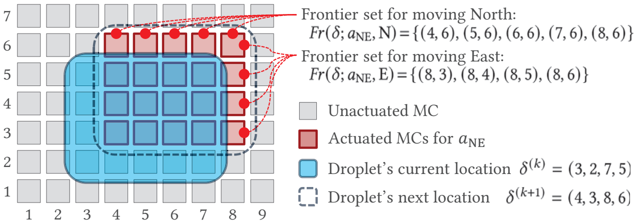
7
6
5
4
3
2
1
1
2
3
4
5
6
7
8
9
Frontier set for moving North:
$Fr(\delta; a_{\mathrm{NE}}, \mathrm{N}) = \{(4,6), (5,6), (6,6), (7,6), (8,6)\}$
Frontier set for moving East:
$Fr(\delta; a_{\mathrm{NE}}, \mathrm{E}) = \{(8,3), (8,4), (8,5), (8,6)\}$
Unactuated MC
Actuated MCs for a_{NE}
Droplet's current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

Fig. 4: MEDA segment with droplet δ actuated under action a_{NE} at time k.

C. 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 $\mathcal{A} = \mathcal{A}_c \cup \mathcal{A}_o \cup \mathcal{A}_m$ (i.e., $\mathcal{A} \subset \{a \mid a : \Delta \to \Delta\}$); here, $\mathcal{A}_c = \{a_{\mathsf{N}}, a_{\mathsf{S}}, a_{\mathsf{E}}, a_{\mathsf{W}}\}$ are movements in the cardinal directions, $\mathcal{A}_o = \{a_{\mathsf{NE}}, a_{\mathsf{NW}}, a_{\mathsf{SE}}, a_{\mathsf{SW}}\}$ are movements in the ordinal directions, and $\mathcal{A}_m = \{a_{\uparrow}, a_{\downarrow}\}$ are morphing transformations that increase or decrease the droplet's aspect ratio, respectively.

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 their respective transformations and frontier sets. For example, Fig. 4 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 $Fr(\delta; a_{\mathsf{NE}}, \mathsf{E}) = [\![(8, 3), (8, 6)]\!]$ and $Fr(\delta; a_{\mathsf{NE}}, \mathsf{N}) = [\![(4, 6), (8, 6)]\!]$, respectively (see Fig. 4).

The degradation level of the MCs used in the movement (i.e., in the frontier set) impacts the EWOD driving force. Thus, a microfluidic action may not always result in the intended droplet movement. From (2), the relative EWOD force exerted on δ in direction d by action a can be estimated as

$$\bar{F}(\delta; a, d) = \sum\nolimits_{Fr(\delta; a, d)} \bar{F}_{ij} \quad = \sum\nolimits_{Fr(\delta; a, d)} \tau^{2n_{ij}/c}.$$

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 $Fr(\delta; a, d)$. Assuming that all MCs in $Fr(\delta; a, d)$ equally contribute to the movement, the probability that droplet δ moves in direction d as a result of executing a, denoted by $p(\delta; a, d)$, can be expressed as

$$p(\delta; a, d) = \begin{cases} \bar{F}(\delta; a, d) / |Fr(\delta; a, d)| & \text{if } Fr(\delta; a, d) \neq \varnothing, \\ 0 & \text{otherwise.} \end{cases}$$

TABLE II: Microfluidic actions, transformations and frontier sets, where $\delta = (x_a, y_a, x_b, y_b)$ and, e.g., $x_a^+ = x_a + 1$ and $x_a^- = x_a - 1$.

$a \in \mathcal{A}$	$a(\delta) \in \Delta$	$Fr(\delta; a, d), d \in \{\mathsf{E}, \mathsf{W}\}$	$Fr(\delta; a, d), d \in \{\mathsf{N}, \mathsf{S}\}$
a_N	(x_a, y_a^+, x_b, y_b^+)	$\varnothing$	$[\![(x_a, y_b^+), (x_b, y_b^+)]\!]$
a_S	(x_a, y_a^-, x_b, y_b^-)	$\varnothing$	$[\![(x_a, y_a^-), (x_b, y_a^-)]\!]$
a_E	(x_a^+, y_a, x_b^+, y_b)	$[\![(x_b^+, y_a), (x_b^+, y_b)]\!]$	$\varnothing$
a_W	(x_a^-, y_a, x_b^-, y_b)	$[\![(x_a^-, y_a), (x_a^-, y_b)]\!]$	$\varnothing$
a_NE	$(x_a^+, y_a^+, x_b^+, y_b^+)$	$[\![(x_b^+, y_a^+), (x_b^+, y_b^+)]\!]$	$[\![(x_a^+, y_b^+), (x_b^+, y_b^+)]\!]$
a_NW	$(x_a^-, y_a^+, x_b^-, y_b^+)$	$[\![(x_a^-, y_a^+), (x_a^-, y_b^+)]\!]$	$[\![(x_a^-, y_b^+), (x_b^-, y_b^+)]\!]$
a_SE	$(x_a^+, y_a^-, x_b^+, y_b^-)$	$[\![(x_b^+, y_a^-), (x_b^+, y_b^-)]\!]$	$[\![(x_a^+, y_a^-), (x_b^+, y_a^-)]\!]$
a_SW	$(x_a^-, y_a^-, x_b^-, y_b^-)$	$[\![(x_a^-, y_a^-), (x_a^-, y_b^-)]\!]$	$[\![(x_a^-, y_a^-), (x_b^-, y_a^-)]\!]$
$a_\uparrow$	(x_a, y_a, x_b^+, y_b^-)	$[\![(x_b^+, y_a), (x_b^+, y_b^-)]\!]$	$\varnothing$
$a_\downarrow$	(x_a, y_a, x_b^-, y_b^+)	$\varnothing$	$[\![(x_a, y_b^+), (x_b^-, y_b^+)]\!]$

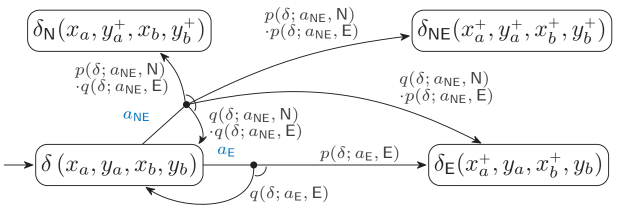
$\delta_\mathsf{N}(x_a, y_a^+, x_b, y_b^+)$
$p(\delta; a_\mathsf{NE}, \mathsf{N})$
$\cdot p(\delta; a_\mathsf{NE}, \mathsf{E})$
$\delta_\mathsf{NE}(x_a^+, y_a^+, x_b^+, y_b^+)$
$p(\delta; a_\mathsf{NE}, \mathsf{N})$
$\cdot q(\delta; a_\mathsf{NE}, \mathsf{E})$
$q(\delta; a_\mathsf{NE}, \mathsf{N})$
$\cdot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

Fig. 5: Model for droplet $\delta = (x_a, y_a, x_b, y_b)$ showing two microfluidic actions a_E and a_NE and their probabilistic transitions, where $q = 1 - p$.

If a is successfully executed on $\delta^{(k)}$, the resulting droplet location is $\delta^{(k+1)} = a\left(\delta^{(k)}\right)$. Otherwise, the droplet location remains unchanged. Fig. 5 shows part of the droplet model for actions a_N and a_NE and their probabilistic transitions.

Consequently, while a droplet δ can be manipulated via various microfluidic actions, 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 utilize stochastic-multiplayer games (SMGs) to model the MEDA biochip. Intuitively, in the MEDA SMG, denoted by $\mathcal{G}$, a droplet controller constitutes the first player, controlling δ, whereas the biochip degradation constitutes the second player, controlling $\mathbf{H}$. The SMG comprises a state-space S that captures all the possible values of δ and $\mathbf{H}$, the set of actions $\mathcal{A}$, and a transition function $\gamma\colon \Delta \times \mathcal{A} \times \Delta \to [0, 1]$, where $\gamma(\delta, a, \delta')$ is the probability that $\delta' = a(\delta)$. For example, in Fig. 5 we have $\gamma(\delta, a_\mathsf{E}, \delta_\mathsf{E}) = p(\delta; a_\mathsf{E}, \mathsf{E})$. Finally, we use $\mathcal{G}$ to synthesize adaptive routing strategies as presented in the next section.

IV. Synthesis Framework

For a given bioassay, the sequencing graph (SG) is preprocessed by a planner that determines the dependencies and module placements of all microfluidic operations (MOs), resulting in a list MOs (e.g., see [14]). 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 loc is the location where the MO is executed. Table III shows the list of MO types and the associated number of input and output droplets.

To synthesize routing strategies, we design a *RJ handler* that decomposes a given MO into a number of droplets *routing jobs* (RJs), each of which 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. Algorithm 1 summarizes the procedure that the handler follows to convert a given MO into a set of single-droplet routing jobs (RJ).

TABLE III: MO types and the number of droplets (input, output) and RJs.

MO.type	Description	#Droplets	#RJs
`dis`	Dispense a droplet (enter biochip)	$(0, 1)$	0
`out,dsc`	Output/Discard a droplet (exit biochip)	$(1, 0)$	1
`mix`	Mix two droplets	$(2, 1)$	2
`spt`	Split a droplet into two	$(1, 2)$	2
`dlt`	Dilute a droplet using another	$(2, 2)$	4
`mag`	Magnetic-sense a droplet	$(1, 1)$	1

Algorithm 1: Handler Procedure

```
Function HANDLE(Microfluidic operation $\mathsf{MO} = (\mathsf{type}, \mathsf{pre}, \mathsf{loc})$)
    Result: Routing job list $(\mathsf{RJ})$
    switch $\mathsf{MO.type}$ do
        case dis do $\mathsf{MO.RJ}[0] \leftarrow (\mathbf{0}, \mathsf{loc}[0], \mathsf{loc}[0])$
        case out, dsc, mag do $\mathsf{MO.RJ}[0] \leftarrow (\mathsf{pre}[0], \mathsf{loc}[0], \mathsf{loc}[0])$
        case mix do
            $\mathsf{MO.RJ}[0] \leftarrow (\mathsf{pre}[0], \mathsf{loc}[0], \textsc{Zone}(\mathsf{pre}[0], \mathsf{loc}[0]))$
            $\mathsf{MO.RJ}[1] \leftarrow (\mathsf{pre}[1], \mathsf{loc}[0], \textsc{Zone}(\mathsf{pre}[1], \mathsf{loc}[0]))$
        case spt do
            $\mathsf{MO.RJ}[0] \leftarrow (\mathsf{pre}[0], \mathsf{loc}[0], \textsc{Zone}(\mathsf{pre}[0], \mathsf{loc}[0]))$
            $\mathsf{MO.RJ}[1] \leftarrow (\mathsf{pre}[0], \mathsf{loc}[1], \textsc{Zone}(\mathsf{pre}[0], \mathsf{loc}[1]))$
        case dlt do
            $\mathsf{MO}_0 \leftarrow (\mathtt{mix}, \mathsf{pre}, \mathsf{loc})$;  $\mathsf{MO}_0.\mathsf{RJ} \leftarrow \textsc{Handle}(\mathsf{MO}_0)$
            $\mathsf{MO}_1 \leftarrow (\mathtt{spt}, \mathsf{MO}_0, \mathsf{loc})$;  $\mathsf{MO}_1.\mathsf{RJ} \leftarrow \textsc{Handle}(\mathsf{MO}_1)$
```

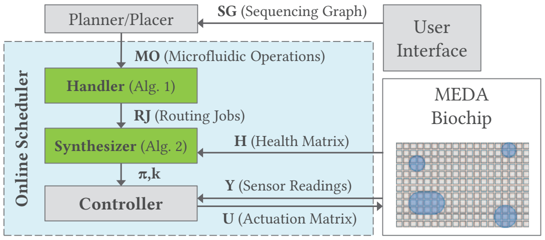

Fig. 6: Synthesis framework data-flow.

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 III. Next, the procedure defines the labels for the goal states g and hazard states h using the δ_g and δ_h of the RJ, respectively. The synthesis requirement is then formalized as the temporal query $\phi\colon \mathsf{R}_{\min=?}\,[\Box\,(\neg h)\,\Diamond g]$, which may be interpreted as "*minimize the number of cycles required to eventually reach a goal state (in g), while never encountering a hazard state (in h)*". Finally, both $\mathcal{G}$ and ϕ are passed to a model checker (e.g., PRISM-games [31]) to synthesize an optimal routing strategy π that maps a droplet location $\delta \in \Delta$ to an optimal action $a \in \mathcal{A}$ (i.e., the strategy is a function $\pi : \Delta \to \mathcal{A}$). The computational complexity of the synthesizer is $\mathcal{O}\left((\mathsf{W}-x)\cdot(\mathsf{H}-y)\cdot|\mathcal{A}|\right)$, where $x \times y$ is the droplet size.

Fig. 6 shows the overall data-flow diagram. The planner provides the list of MOs to the handler 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.

Algorithm 2: Routing Strategy Synthesis Procedure

```
Function $\textsc{Synth}(\mathsf{RJ} = (\delta_s, \delta_g, \delta_h), \textit{health matrix } \mathbf{H})$
    Result: Strategy $\pi : \Delta \rightarrow \mathcal{A}$; expected completion time $k \in \mathbb{R}$
    Initialize model $\mathcal{G}$ using $\mathbf{H}$
    foreach $a \in \mathcal{A}$ do add $\gamma(\delta, a, \delta')$ to $\mathcal{G}$ based on Table II
    Let $g(\delta)$: $(x_a \geq x_a^g) \wedge (y_a \geq y_a^g) \wedge (x_b \leq x_b^g) \wedge (y_b \leq y_b^g)$
    Let $h(\delta)$: $(x_a < x_a^h) \vee (y_a < y_a^h) \vee (x_b > x_b^h) \vee (y_b > y_b^h)$
    Let $\phi$: $\mathsf{R}_{\min=?} [\Box (\neg h) \Diamond g]$; $(\pi, k) \leftarrow \textsc{PrismG}(\mathcal{G}, \phi, \delta_s)$
```

Algorithm 3: Online Scheduler

```
Input: Microfluidic operations list $(\mathsf{MO})$, $k = 0$
foreach MO do  $\mathsf{MO.RJ} \leftarrow \textsc{Handle}(\mathsf{MO})$, $\mathsf{MO.state} \leftarrow \texttt{init}$
while $\exists i : \mathsf{MO}[i]\mathsf{.state} \neq \textit{done}$ do $\mathbf{U} \leftarrow \mathbf{0}$;  Read $\mathbf{Y}$
    foreach $\mathsf{MO} \in (\mathsf{MO})$ do switch $\mathsf{MO.state}$
        case $\texttt{init}$ do
            if $\forall j : \mathsf{MO.pre}[j]\mathsf{.state} == \textit{done}$ then
                $\mathsf{MO.state} \leftarrow \texttt{active}$;  Read $\mathbf{H}^{(k)}$
                $\forall j : \mathsf{MO.RJ}[j].(\pi, k) \leftarrow \textsc{Synth}(\mathsf{RJ}[j], \mathbf{H}^{(k)})$
        case $\texttt{active}$ do
            if $\forall j : \mathsf{MO.RJ}[j].\delta_g == \delta$ then $\mathsf{MO.state} \leftarrow \texttt{done}$
            else $a \leftarrow \pi(\delta)$;  $\mathbf{U}(a(\delta)) \leftarrow \mathbf{1}$
    Apply $\mathbf{U}$;  $k$++
```

Algorithm 3 summarizes the online scheduling procedure for a bioassay. First, the handler populates the list of routing jobs. Next, the scheduler checks whether any MOs are ready for execution and, if any, reads the current health matrix $\mathbf{H}^{(k)}$ and forwards the corresponding RJs to the synthesizer to obtain their routing strategies π. For an active MO, the optimal action a is retrieved from the current strategy (i.e., $a = \pi(\delta)$), and the corresponding MCs are 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.

V. Experimental Evaluation

We implemented the online scheduler, handler, synthesizer, and a MEDA biochip simulator in MATLAB. The synthesizer automatically generated and passed routing jobs to PRISM-games [31] to obtain routing strategies. Results were obtained on an Intel Core i7 2.6 GHz CPU with 16 GB RAM. We simulated four benchmark bioassays in our experiments: Master-Mix, CEP, Serial Dilution [32], and nucleosome immunoprecipitation (NuIP) [15]. NuIP is used for studying the epigenetic relationship between DNA and its supporting proteins [33].

We first simulated a fabricated MEDA biochip with 30×60 MCs [7]. Each MC followed the reliability model in (1), with degradation constants $c \in [200, 500]$ and $\tau \in [0.5, 0.9]$, uniformly sampled within their corresponding intervals to simulate microelectrode degradation. Once assigned, both c and τ remained constant during each set of experiments. Two routing algorithms were implemented; the first algorithm (baseline) is unaware of degradation and finds the shortest path to minimize bioassay completion time, while the second synthesized adaptive routing strategies based on the proposed framework. Neither approach utilized the error-recovery techniques described in [34], [35] as our goal is to proactively avoid errors and the cost associated with error recovery.

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 ($k_{\max}$). Fig. 7 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.

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 III, 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. 8 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.

VI. 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 presented a stochastic game-based

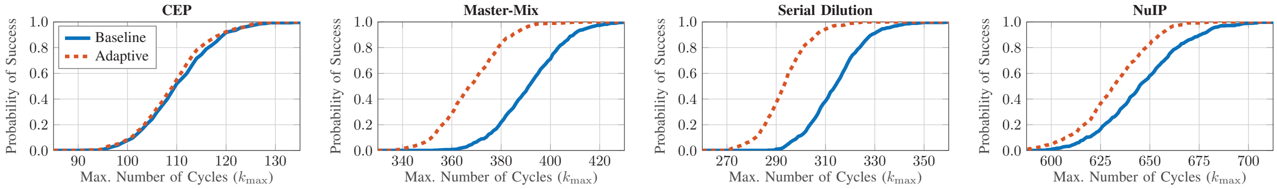

Fig. 7: Probability of successful bioassay completion versus the maximum number of cycles.

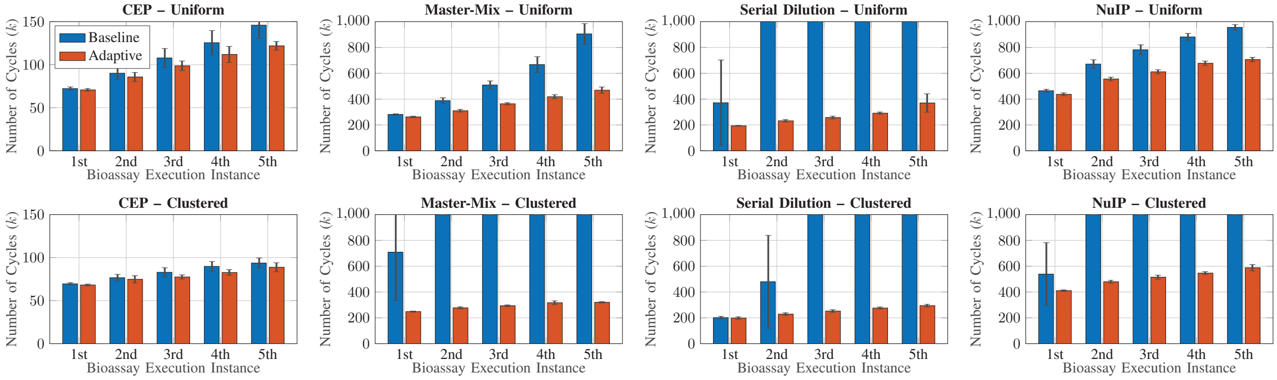

Fig. 8: Average number of cycles required to execute a bioassay under different routing strategies and fault-injection modes (standard deviations are indicated).

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.

References

[1] W.-L. Chou *et al.*, "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 *et al.*, "Digital microfluidics for the detection of selected inorganic ions in aerosols," *Sensors*, vol. 20, no.5, p.1281, 2020.

[4] A. Ganguli *et al.*, "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] 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.

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

[9] Y. Ho *et al.*, "Design of a micro-electrode cell for programmable lab-on-cmos platform," in *Proc. ISCAS*, 2016, pp.2871–2874.

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

[11] 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.

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

[13] 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.

[14] Z. Zhong *et al.*, "Micro-electrode-dot-array digital microfluidic biochips: Technology, design automation, and test techniques," *IEEE TBioCAS*, vol. 13, no.2, pp.292–313, 2018.

[15] T.-C. Liang *et al.*, "Adaptive droplet routing in digital microfluidic biochips using deep reinforcement learning," in *Proc. ICML*, 2020.

[16] O. Keszocze *et al.*, "Exact one-pass synthesis of digital microfluidic biochips," in *Proc. DAC*, 2014, pp.1–6.

[17] 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.

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

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

[20] O. Keszocze *et al.*, "A general and exact routing methodology for digital microfluidic biochips," in *Proc. ICCAD*, 2015, pp.874–881.

[21] Q. Wang *et al.*, "Control-fluidic codesign for paper-based digital microfluidic biochips," in *Proc. ICCAD*, 2016, pp.1–8.

[22] J. McDaniel *et al.*, "PCB escape routing and layer minimization for digital microfluidic biochips," *IEEE TCAD*, vol. 36, no. 1, pp. 69–82, 2016.

[23] O. Keszocze *et al.*, "Exact routing for micro-electrode-dot-array digital microfluidic biochips," in *Proc. ASP-DAC*, 2017, pp.708–713.

[24] S. Zafar *et al.*, "Charge trapping related threshold voltage instabilities in high permittivity gate dielectric stacks," *Journal of Applied Physics*, vol. 93, no.11, pp.9298–9303, 2003.

[25] C.-H. Lee *et al.*, "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.

[26] C. Besset *et al.*, "MIM capacitance variation under electrical stress," *Microelectronics Reliability*, vol. 43, no.8, pp.1237–1240, 2003.

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

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

[29] Y. Liu *et al.*, "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.

[30] C. Dong *et al.*, "On the droplet velocity and electrode lifetime of digital microfluidics: voltage actuation techniques and comparison," *Microfluidics and Nanofluidics*, vol. 18, pp.673–683, 2015.

[31] M. Kwiatkowska *et al.*, "Prism-games 3.0: Stochastic game verification with concurrency, equilibria and time," in *Proc. CAV*, 2020, pp.475–487.

[32] M. Elfar *et al.*, "Synthesis of error-recovery protocols for micro-electrode-dot-array digital microfluidic biochips," *ACM TECS*, vol. 16, pp.1–22, 2017.

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

[34] Z. Li *et al.*, "Efficient and adaptive error recovery in a micro-electrode-dot-array digital microfluidic biochip," *IEEE TCAD*, vol.37, pp.601–614, 2017.

[35] 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.