Error Recovery in Cyberphysical Digital-Microfluidic Biochips

Yan Luo†, Krishnendu Chakrabarty†, and Tsung-Yi Ho‡

†Electrical & Computer Engineering Dept.
Duke University, Durham, NC 27708, USA

‡Computer Science and Information Engineering Dept.
National Cheng Kung University, Tainan, Taiwan

***Abstract*—Droplet-based "digital" microfluidics technology has now come of age and software-controlled biochips for healthcare applications are starting to emerge. However, today's digital microfluidic biochips suffer from the drawback that there is no feedback to the control software from the underlying hardware platform. Due to the lack of precision inherent in biochemical experiments, errors are likely during droplet manipulation, but error recovery based on the repetition of experiments leads to wastage of expensive reagents and hard-to-prepare samples. By exploiting recent advances in the integration of optical detectors (sensors) in a digital microfluidics biochip, we present a "physical-aware" system reconfiguration technique that uses sensor data at intermediate checkpoints to dynamically reconfigure the biochip. A cyberphysical re-synthesis technique is used to recompute electrode-actuation sequences, thereby deriving new schedules, module placement, and droplet routing pathways, with minimum impact on the time-to-response.**

I. Introduction

MICROFLUIDIC biochips have now come of age, with applications to biomolecular recognition for high-throughput DNA sequencing, immunoassays, and point-of-care clinical diagnostics [1]. In particular, digital microfluidic biochips, which use electrowetting-on-dielectric to manipulate discrete droplets (or "packets of biochemical payload") of picoliter volumes under clock control, are especially promising [2].

The ease of reconfigurabilty and software-based control in digital microfluidics has motivated research on various aspects of automated chip design and chip application. A number of techniques have been published for architectural-level synthesis [3], module placement, and droplet routing [4], [5], [6]. However, these techniques ignore domain-specific constraints or practical realities that arise from attempting to carry out biochemical reactions and microfluidic operations on an electronic chip. Due to the randomness and complex component interactions that are ubiquitous in biological/chemical processes, predictive modeling and accuracy control are difficult [7], [8]. In addition to manufacturing defects and imperfections, faults may also arise during bioassay execution. For example, excessive actuation voltage applied to an electrode may lead to breakdown of electrodes and charge trapping, and DNA fouling may lead to malfunction of multiple electrodes in the biochip [9] [10] [11]. These faults are hard to detect *a priori*, but they occur often during bioasays [11]. Yet, despite such inherent variability, many biomedical applications such as drug development and clinical diagnostics require high precision for each operation and correctness of the final result under various conditions. If an unexpected error occurs during the experiment, the outcome of the entire experiment will be incorrect. As a result, all the steps in the experiment must be repeated in order to correct the error [12].

The repetitive execution of on-chip laboratory experiments leads to the following problems: (i) wastage of samples that are difficult to obtain or prepare, and the wastage of expensive reagents; (ii) an increase in the time-to-result for a bioassay, which is detrimental to real-time detection and rapid response. Therefore, it is necessary to develop techniques for monitoring assay outcomes at intermediate stages and design an efficient error-recovery mechanism.

Error recovery in digital microfluidics has received relatively little attention in the literature. The only reported work is [13], which proposed intermediate stage monitoring and rollback error-recovery for a microfluidic biochip. The key idea in this work is to use optical sensors to verify the correctness of immediate product droplets at various steps in the on-chip experiment. Optical detection has been integrated with digital microfluidics to evaluate the concentration and volume of product droplets [2], [14], [15]. In the recent approach described in [13], error recovery is carried out as follows. During bioassay execution, intermediate product droplets are sent to optical sensors. When an error is detected at an optical sensor, i.e., the volume or concentration of the droplet is below or above the acceptable calibrated range, the corresponding droplet is discarded. The operations whose outputs fail to meet the quality requirements based on sensor calibration are re-executed to generate a new product droplet to replace the unqualified droplet.

Figure 1 shows an example of rollback error-recovery. The initial sequencing graph of a bioassay is shown in Figure 1(a). Here we assume that the outputs of each dispensing, mixing and splitting operation are evaluated by an optical sensor. When an error occurs at operation 9, the system will re-execute the corresponding dispensing and mixing operations. The new sequencing graph for error recovery is shown in Figure 1(b). Operations 16, 17, 18, and 19 are added for error recovery. The new sequencing graph for error recovery is shown in Figure 1(b). Operations 16, 17, 18, and 19 are added for error recovery. It is important to note that, the error is detected at operation 9, thus it is unnecessary to continue executing the splitting operation after operation 9 anymore. Thus node 12 is removed from Figure 1(b).

In the absence of "physical-aware" control software, the error recovery method in [13] suffers from following drawbacks:

The work of Y. Luo and K. Chakrabarty was supported in part by the US National Science Foundation under grants CCF-0914895 and CNS-1135853. The work of T.-Y. Ho was supported in part by the Taiwan National Science Council under grant NSC 100-2220-E-006-016.

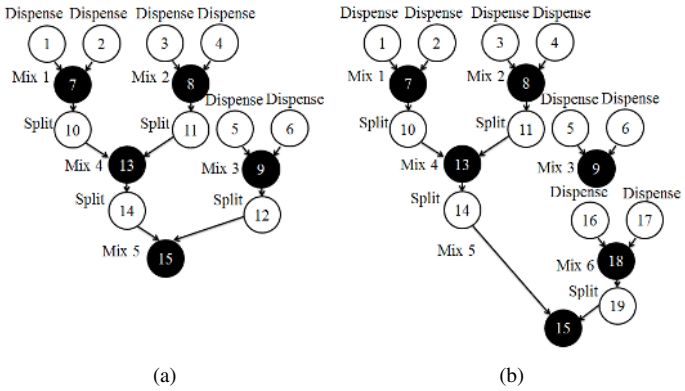


Fig. 1. (a) Initial sequencing graph; (b) operations 16, 17, 18 and 19 are added for error recovery.

1) The first drawback is the over-simplification of fault detection and the associated assumptions. Using a uniform "expected value" for the calibration of each optical detection operation is not practical. Note that at various stages during bioassay execution, the concentration of intermediate product droplets vary in a dynamic manner, hence the calibration also needs to be repeated and carried out dynamically.
2) In [13], all recovery operations are carried out in a stand-alone manner. When an error is detected, all other ongoing bioassay-related fluidic operations are interrupted. The potential long waiting times introduced by recovery operations can lead to sample degradation and erroneous assay outcomes [16]. Some operations, such as colorimetric enzyme-kinetic reactions, require precise durations as specified by the reaction protocol, and they cannot be extended without introducing unpredictability in the experiment outcome [15].
3) The error recovery approach in [13] cannot handle situations when multiple errors occur during a bioassay. For example, it assumes that all error recovery operations will be executed successfully and it does not consider the likelihood that errors can also occur during recovery.
4) The error recovery strategy in [13] does not consider reliability issues. Errors such as the generation of droplets with abnormal volumes are usually caused by the accumulation of charge on the surface of certain electrodes [9] [10]. If the use of such electrodes is continued, it is likely that they will introduce more errors [9] [10]. Thus, in order to ensure the reliability of biochips, we must minimize the utilization of these electrodes.

To overcome the above drawbacks, we take a transformative "cyberphysical" approach towards achieving closed-loop and sensor feedback-driven biochip operation under program control. By exploiting recent advances in the integration of optical detectors (sensors) in a digital microfluidics biochip [14], we present a "physical-aware" system reconfiguration technique that uses sensor data at intermediate checkpoints to dynamically reconfigure the biochip [17].

The key contributions of this paper are as follows:

- Charge-coupled device (CCD)-based and optical detector-based sensing systems for digital microfluidic biochip are proposed and compared (Section II).
- An algorithm is proposed for the measurement and tracking of droplets based on real-time imaging data from a CCD camera (Section II).
- A reliability-driven error recovery strategy is proposed (Section III).
- Parallel recombinative simulated annealing (PRSA)-based and greedy algorithms for reliability-driven synthesis are presented (Section IV).
- Simulation results are presented for three representative bioassays (Section V).

II. Overview of Digital Microfluidic Biochips And Cyberphysical System

A digital microfluidic system includes a microfluidic biochip and control software. A digital microfluidic biochip consists of three types of components, i.e., a two-dimensional electrode array, on-chip reservoirs, and sensors [18]. By utilizing the principle of electrowetting, picoliter droplets containing biological/chemical samples and reagents can be manipulated on the chip without any curved channels or external pressure sources [19] [2].

Figure 2 shows the structure of an element cell on a microfluidic biochip [2]. The upper plate is a large electrode that covers all cells on the array and serves as the ground electrode for all unit cells [18] [2]. The lower plate of the unit cell consists of an array of discrete control electrodes. During biochip operation, the unit cells in the array may have different voltages applied on their lower electrodes [18] [2]. The movements of droplets are determined by signals applied on the discrete lower electrodes of the array.

As seen in Figure 2, droplets manipulated by the digital microfluidic biochip are confined between the upper and lower electrodes [2]. To move a droplet, a high voltage must be applied to a unit cell adjacent to the droplet, and, at the same time, a low voltage must be applied to the cell under the droplet [2]. The voltages applied on the electrodes can influence the surface characteristics of the hydrophobic material on the lower electrodes. In this way, the different voltages applied on the electrodes result in different levels of interfacial tension on the surface of the biochip [2]. Due to this effect, the droplet is moved from the low-voltage electrode to the high-voltage electrode [2]. All the microfluidic functions, such as droplet merging, splitting, mixing, and dispensing can be reduced to a set of basic operations [18]. Concurrent manipulation of multiple discrete droplets can be coordinated by control software and by voltages applied to the electrodes, so there is no need for mechanical devices, such as tubes, pumps, and valves [19].

With the availability of sensing system for biochips, "physical-aware" control software becomes feasible. By "physical-aware", we refer to the fact that the software can receive information about the outcome (error-free/erroneous) of fluid-handling operations based on feedback from the sensing system. Depending on sensor feedback, the control software can appropriately reconfigure the microfluidic biochip. In this way, the various steps in the bioassay are executed based upon real-time sensing of intermediate results.

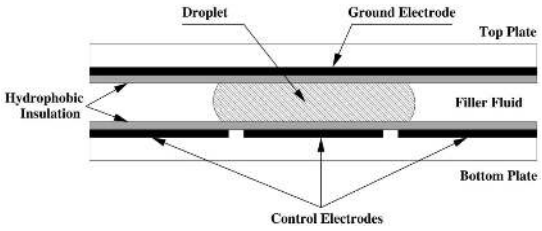


Fig. 2. Schematic cross section of a unit cell on digital microfluidic arrays [2].

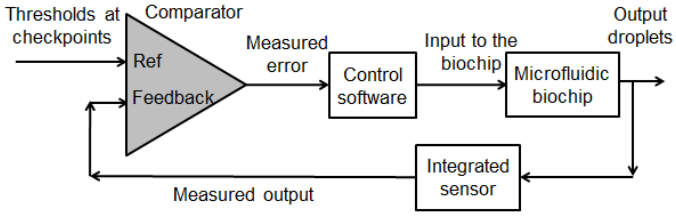


Fig. 3. The schematic of the cyberphysical digital microfluidic system.

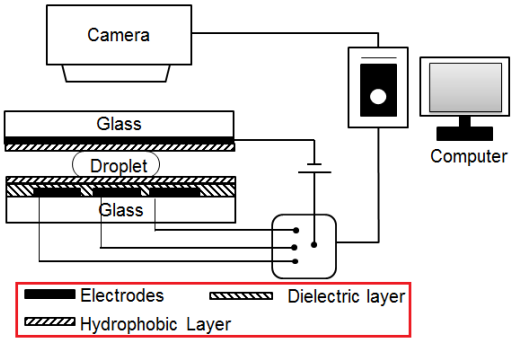


Fig. 4. Illustration of a digital microfluidics-based image processing system [20].

Figure 3 depicts each component of a cyberphysical system on the microfluidic platform. The control software sends a control signal to the microfluidic biochip, and the on-chip sensing system monitors the outcomes of fluidic operations. The outcomes are compared with the "expected values", i.e., the pre-determined thresholds. If the results of the comparison indicate that an error has occurred, the control software receives a "repeat request", and the corresponding operation in which the error occurred can be executed again, thereby correcting the error.

A. Sensing Systems

Two sensing schemes can potentially be used in the cyber-phyiscal system on digital microfluidic biochips.

The first sensing scheme is CCD camera-based. As described in [2] [20], CCD cameras can be used in experiments to view the top sides of droplets simultaneously. An example of the CCD monitoring system is shown in Figure 4.

Based on images captured by the CCD camera, droplets can be automatically located by the control software. The procedure of automatically searching for droplets can be described as a "template matching" problem. Here a pattern can be represented as the image of a "typical" droplet. During the matching process, we move the template image to all possible positions in the image of the entire array and crop a sub-image that has the same size as the template image. Then the control software computes the correlation index, which indicates the similarity between the template and the "cropped image". This process is shown in Figure 5(a) and the correlation factor is calculated on a pixel-by-pixel basis.

In the control software, all images are stored in grayscale form, which can be encoded as matrices or vectors. Suppose the template image is represented in a 1-D array: $\vec{x} = (x_1, x_2, ...x_N)$. Here x_i represents the gravel level of a pixel and N is the total number of pixels in the template image. Similarly, the cropped sub-image to be compared with the template image can be written as $\vec{y} = (y_1, y_2, ...y_N)$. Thus the correlation factor between these two images is defined as:

$$cor = \frac{\sum_{i=1}^{N} (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N} (x_i - \bar{x})^2 \cdot \sum_{i=1}^{N} (y_i - \bar{y})^2}},$$

where $\bar{x}$ and $\bar{y}$ are the average gray level in the template image and cropped sub-image, respectively. The range of correlation factor cor is a real number between -1 and $+1$. According to the definition of correlation, a larger value represents a stronger relationship between two images.

After deriving the correlation factors for all possible positions in the image for the complete biochip, we obtain the correlation map between the template and the original input image. Suppose there are κ droplets on the biochip. By searching for the largest κ correlation factors in the correlation map, the locations of droplets can be determined. An example is shown in Figure 5(b) and (c) [21]. Part (b) shows the original input image of the whole chip and the pattern image, and (c) is the correlation map, where the best matching locations, i.e. the coordinations of droplets derived by the control software are (77, 107), (77, 147) and (76, 208). Thus the control software automatically locates the droplets, and it can further analyze the sizes and colors of droplets according to the image. In this manner, the volumes and concentrations of droplets can be acquired after processing the image taken by the CCD camera.

Instead of searching for droplets in the complete image, we use imaging to check whether the droplets have been moved to the expected positions. This is implemented using the following steps:

First, we do some calibration before the experiment. We choose a large number of sub-images with (or without) droplets, and calculate their correlation with the template. Based on this calculation, we find an appropriate threshold for the correlation index (C_{th}): if the correlation is larger than C_{th}, we conclude that there is a droplet in the cropped sub-image; otherwise, there is no droplet in the sub-image. When the bioassay is running, we only need to crop the sub-images near the expected positions of droplets, and calculate their corresponding correlation indices to determine the absence/presence of droplets.

The advantages of the CCD camera-based sensing system are: (i) the detection of errors immediately after they occur and (ii) the identification of the precise locations of the errors.

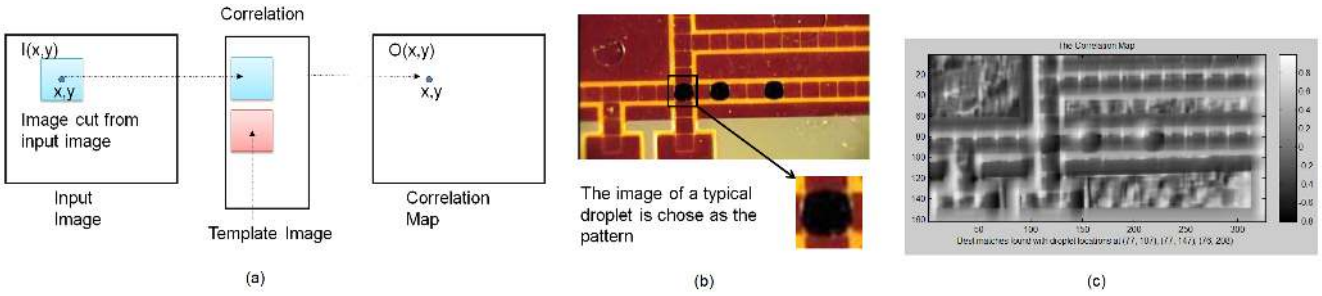


Fig. 5. (a) The matching process moves the template image to all possible positions in a larger source image and computes a numerical index that indicates how well the template matches the sub-image in that position; (b) the image of the whole biochip [21] and the pattern we selected; (c) the correlation map between image of the whole array and the pattern. The positions of droplets can be determined by finding κ maximum elements (κ is the number of droplets on the chip) in the correlation map.

A disadvantage of this system is that extra instruments, such as CCD cameras, are required to observe the cyberphysical system.

The second sensing scheme is based on integrated optical detectors, as proposed in [14] [15]. The quality of an intermediate product in a digital microfluidic biochip can be determined by examining the concentration of the product in the droplets through fluorescence [14] [15]. When a fluorophore tag is attached to a droplet, different product concentrations lead to the emission of light different wavelengths (i.e., different colors). This difference in color can be detected by optical sensors that convert the received light into electrical current or a voltage signal [14]. In recent work, integrated photodetectors have been introduced on the microfluidic array [14] [15]. For example, in [15], an optical detection system was integrated with the digital microfluidic array. It consists of a light-emitting diode (LED) and a photodiode which functions as light-to-voltage converter. The concentration of products on the array can be measured by the absorbance of the droplets using a kinetic rate method. As another example, thin film InGaAs photodetectors can be bonded onto a glass platform, coated with Teflon AF, and then integrated into the digital microfluidic system. Figure 6 shows a coplanar digital microfluidic chip with the integrated InGaAs photodetector [15].

Even though no instruments with large footprint and precise alignment are needed in this method, the integrated optical detector-based sensing system has a drawback that it cannot precisely locate the electrode where an error has occurred. For example, when an output droplet of an operation is sent to the detector and it fails to meet the requirement of the bioassay, we know that an error occurred during the mixing operation, but we cannot locate the precise time and the position where it occurred.

B. "Physical-Aware" Software

The availability of on-chip sensors provides digital microfluidic biochips with the capability of using sensor data at intermediate checkpoints to detect errors, thereby minimizing the impact of errors that occur during bioassay execution. The work in [13] proposed intermediate stage monitoring and rollback error-recovery for a microfluidic biochip. The key idea in this work is using the sensing system on-chip to verify

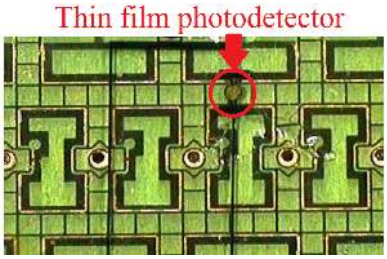


Fig. 6. Coplanar digital microfluidic chip with integrated thin film photodetector [15].

the correctness of output droplets at various steps in the on-chip experiment. In this approach, error recovery is carried out as follows. When an error is detected at a checkpoint, operations whose outputs failed to meet the quality requirements based on sensor calibration are re-executed to recover from the error. Additional intermediate product droplets must be stored in specially designated locations of the chips to facilitate recovery. Additional droplets of samples and reagents must also be dispensed from reservoirs for error recovery. The details of the strategy for reliability-driven error recovery are presented in Section III, and the algorithm for dynamic re-synthesis of error recovery is described in Section IV.

C. Interfaces between Biochip and Control Software

We next describe the cyberphysical coupling between the control software and the hardware of the microfluidic platform. There are two interfaces needed for cyberphysical coupling. The first interface converts the output signals from the sensing system to the inputs of the desktop computer that the control software can interpret. The second interface converts the output data generated by the control software to voltage signals that can be directly applied to the electrodes of the biochip.

As described in [3], biochip synthesis includes resource binding and scheduling, which specifies the start and stop times of fluidic operations. The synthesis results (control software) need to be mapped to a sequence of electrode actuation vectors consisting of "0", "1", and "F" (floating). A programmable logic controller (PLC) is used as the interface between the output of the control software ("controller") and the control pins of the biochip. The controller transfers data to the memory of the PLC through the USB port of the computer and an RS-232 interface.

During bioassay execution, the PLC reads the electrode actuation sequences stored in memory and applies the appropriate sequence of voltages to the output pins. When the PLC generates output voltages based on sequences in its memory, corresponding activation voltages are applied to the electrodes of the biochip. The quality of an intermediate product in a digital microfluidic biochip can be determined by examining the product concentration in droplets through fluorescence [22], [23]. When a fluorophore tag is attached to a droplet, different product concentrations lead to emitted light of different wavelengths (i.e., different colors). This difference in color can be detected by the sensing system that converts the received analog signals into electrical current or voltage [2]. Recent advances in the integration of a multiplicity of such miniaturized sensors provide an important motivation for the cyberphysical hardware/software co-design approach studied in this paper.

III. Reliability-Driven Error Recovery Strategy

A. Error Recovery Strategies

In this subsection, we formulate the principles underlying error recovery. For the given bioassay protocol, we use the sensing system on-chip to evaluate the quality of output droplets of each dispensing, mixing, dilution and splitting operation. According to the data provided by [2], the response time of on-chip optical sensors are in the scale of picoseconds or nanoseconds. Thus the time cost for adding optical detection operations is negligible.

For a microfluidic biochip, fluid-handling operations can be divided into two categories: reversible and nonreversible operations. Reversible operations include dispensing and splitting operations; nonreversible operations include mixing and dilution operations. For errors that occur at reversible operations, their recovery processes are relative simple. In a splitting operation, if two droplets with unbalanced volumes are generated, then the biochip will first merge the two abnormal droplets to a larger one and then split the larger droplet again. For errors that occur at a dispensing operation, the chip can send the abnormal droplet back to the corresponding reservoir and dispense another droplet. Thus for errors that occur at reversible operations, the time cost for recovery is small and no additional droplets need to be consumed.

The error recovery process for nonreversible operations is more involved. To implement the corresponding nonreversible operations to correct the error, we also need input droplets from operations whose outputs feed the inputs of the failed operation. Thus we may need to re-execute all the predecessors of the erroneous operation. For instance, if an error occurs at operation 7 in Figure 7(a), operations 1, 2, 3, 4, 5, and 6 may need to be re-executed. Thus the time cost for executing error recovery operations can be extremely high. To reduce the incidence of the worst case, the following strategies are taken in our approach:

• For a splitting operation, if only one of its output droplets is used as the input for the immediate successors, we store the other (redundant) droplet as a backup for possible error recovery at a subsequent stage. For example, operation 7 in Figure 7(a) is a splitting operation and it generates two output droplets. Only one of these two droplets is used as the input of operation 9. (Note here each circle in the sequencing graph stands for a fluid-handling operation. The unused droplets are not shown in the sequencing graph.) If an error occurs at operation 9, the redundant droplet will be used as an input for re-execution.

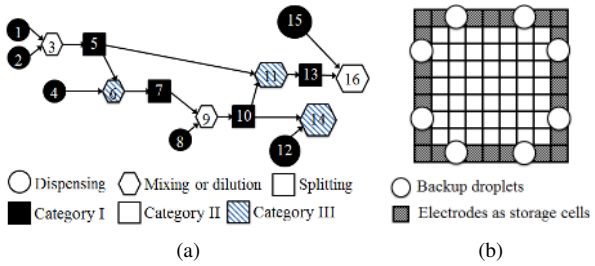


Fig. 7. (a) An example of a sequencing graph corresponding to a bioassay protocol; (b) The layout of a biochip with reserved area for error recovery.

• All dispensing operations are scheduled for execution as early as possible and their output droplets are stored on the biochip. We also dispense some droplets as backup for possible error recovery operations. When the bioassay is completed, those unused backup droplets are sent back to the corresponding reservoirs.

Thus, when an error occurs at a nonreversible operation, the control software first checks whether the inputs of this operation can be provided by backup droplets stored on chip. If the answer is yes, then the time cost for this operation can be reduced. Otherwise, more operations will be executed during error recovery. Based on the above discussion, the operations in the sequencing graph can be divided into three categories according to the number of operations and droplet consumptions in their error recovery processes, as shown in Figure 7(a).

The above operations can be formally categorized as follows:

Category I: This is the set of all reversible operations. They can be simply re-executed when an error occurs.

Category II: This is the set of nonreversible operations for which immediate predecessors can provide backup droplets.

Category III: This corresponds to the set of nonreversible operations for which their immediate predecessors cannot provide backup droplets.

In a given sequencing graph, each node represents an operation. We define the number of input droplets as the in-number of an operation, and the number of output droplets as the out-number of an operation. As described below, any operation opt_k can be categorized based on its in-number and out-number values:

- If in-number of opt_k is equal to zero, then opt_k is a dispensing operation. Thus we have: $opt_k \in$ Category I.
- If in-number of opt_k is equal to one and the out-number of opt_k is equal to two, then opt_k is a splitting operation. Thus we have: $opt_k \in$ Category I.
- Suppose opt_j is an immediate predecessor of opt_k. Then the number of backup droplets at the output of opt_j can be calculated as: $B_{opt_j} = ON_j - MN_j$, where

1: Classify operations into Category I, Category II and Category III;
2: Initialization of $\mathcal{R}_i$: $\mathcal{R}_i = opt_i$;
3: Initialization of intermediate variable Re: $Re = \mathcal{P}_r(\mathcal{R}_i)$;
4: **while** $(Re - \mathcal{R}_i) \cap \{\text{Set of operations in Category III}\} \neq \emptyset$ **do**
5: Update $\mathcal{R}_i$: $\mathcal{R}_i = \mathcal{P}_r(\mathcal{R}_i)$;
6: Update Re: $Re = \mathcal{P}_r(\mathcal{R}_i)$;
7: **end while**
8: $\mathcal{R}_i = Re$;
9: $\mathcal{R}_i$ is the set of recovery operation for opt_i;

Fig. 8. Pseudocode for determining the recovery operation for opt_i.

ON_j is out-number of opt_j, and MN_j is the number of immediate successors of opt_j. If the numbers of backup droplets for opt_k's immediate predecessors are all non-zero, then we have: $opt_k \in$ Category II; otherwise, $opt_k \in$ Category III.

For an operation opt_i, the set of its error recovery operations, $\mathcal{R}_i$, can be derived according to the categorization result for opt_i. Operations in Category I and II can be simply re-executed when an error occurs because their input droplets are stored on chip. However, for operations in Category III, their inputs come from the outputs of predecessor operations and we do not have backup for these droplets. Thus if an error occurs in an operation of Category III, we not only need to re-execute the operation itself but also need to backtrace to its predecessors. Suppose the error operations is opt_e and its immediate predecessors are operation opt_{p_1} and opt_{p_2}. If these immediate predecessors are operations in Category I or Category II, we can first re-execute opt_{p_1}, opt_{p_2} and then opt_e for error recovery, thus $\mathcal{R}_i = \{opt_{p_1}, opt_{p_2}, opt_e\}$. If the immediate predecessors opt_{p_1} and opt_{p_2} are neither in Category I nor Category II, we have to continue enlarging $\mathcal{R}_i$ by adding the immediate predecessors of opt_{p_1} and opt_{p_2} into $\mathcal{R}_i$. This backtracing and enlargement procedure needs to be repeated until we reach predecessor operations that can provide backup droplets to feed the inputs of operations in the set of error operations.

The above procedure of backtracing and enlargement of the set $\mathcal{R}_i$ can be described as follows. First, we define the mapping $pred(opt_i)$ to be a mapping from opt_i to the set of immediate predecessors of opt_i in the sequencing graph.

For a set of operations $\mathcal{O} = \{opt_{o_1}, opt_{o_2}, ...opt_{o_k}\}$, we define the operator $\mathcal{P}_r$ as:

$$\mathcal{P}_r : \mathcal{O} \rightarrow \bigcup_{i=o_1,o_2,...o_k} \{opt_i, opt_j | opt_j \in pred(opt_i), \forall\ j\}$$

where $\mathcal{P}_r$ is a backtracing operation. For any operation opt_i, its set of error recovery operations $\mathcal{R}_i$ can be derived by the procedure presented in Figure 8. According to above discussion, we can derive the set of recovery operations $\mathcal{R}_i$ for any operation opt_i.

Based on the relationship between operations in the initial sequencing graph, we can further add edges between operations in the set $\mathcal{R}_i$, and thus derive the error recovery graph G_{Re_i} for opt_i. If an error occurs in opt_i, we will re-execute operations in G_{Re_i} for error recovery.

It is important to note that some electrodes on the biochip are intentionally left unused and reserved for storage of backup droplets. An example is shown in Figure 7(b); all electrodes on the boundary of the chip are used as storage cells. Thus backup droplets can be easily transported on the biochip.

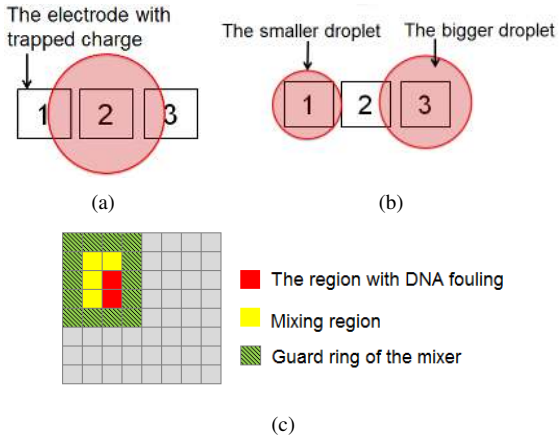


Fig. 9. (a) An error caused by the phenomenon of charge trapping; (b) splitting operation with droplets with unbalanced volumes; (c) an error caused by DNA fouling on the surface of a biochip.

B. Reliability Consideration in Recovery

When an error is detected during the execution of a bioassay, simply re-executing the operation for which an error occurred is not efficient to ensure reliable operation. This is because the errors that occur during the execution of a bioassay usually are caused by defects involving electrodes; thus, multiple errors may occur in the same region of the biochip at different times. Two examples are provided below to illustrate the errors caused by the charge-trapping phenomenon and DNA fouling.

When the electrodes of a digital microfludic biochip are actuated excessively, physically-trapped charge and residual charge may lead to reliability problems [9] [10]. Charge trapping is a phenomenon in which charge is trapped and concentrated in the dielectric insulator of the chip. It can lead to a reduction in the electrowetting force and malfunctions in the execution of the bioassay. An example is shown in Figure 9(a). Suppose Electrode 1 has a trapped charge in its dielectric insulator layer, while Electrode 2 and 3 do not suffer from charge sharing. In order to implement a splitting operation, high voltages are applied on Electrode 1 and Electrode 3. However, the charge trapped on Electrode 1 will reduce the electrowetting force. The droplet will be split by unequal forces, and the two resulting droplets may have unequal volumes; see Figure 9(b). If we simply re-execute the splitting operation and continue to use Electrode 1, additional errors may result. Even worse, the charge-trapping phenomenon eventually may cause permanent dielectric degradation of the electrode [9] [10]. Thus, once an error is detected, the electrode at which the error occurred must no longer be used to implement fluid-handling operations in order to ensure the reliability of the biochip.

When droplets contain macromolecules, such as DNA, they may foul the surface of the electrodes [24]. As a result, droplet concentration can change in undesirable ways. If we continue to use these contaminated electrodes, other droplets may also

TABLE I
SYNTHESIS RESULTS FOR THE BIOASSAY SHOWN IN FIGURE 1(A).

Operation	Start time	Stop time	Resource	Location
Mix 1	6	12	3×2 mixer	(2, 6)
Mix 2	0	6	2×3 mixer	(2, 5)
Mix 3	0	10	2×2 mixer	(6, 2)
Mix 4	12	15	4×4 mixer	(4, 6)
Mix 5	15	18	4×2 mixer	(4, 6)

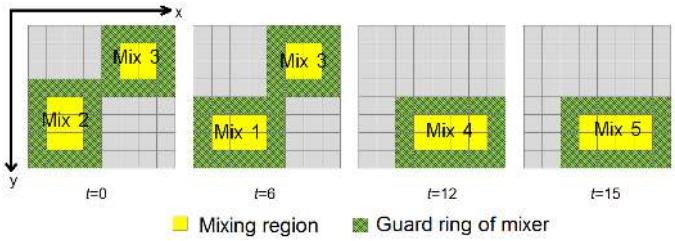


Fig. 10. Module placement results for the bioassay shown in Figure 1(a).

be contaminated. An example of this is shown in Figure 9(c). The region where DNA fouling occurred is used as part of a mixer, and the concentrations of the output droplets of the mixing operation may be abnormal.

We use a simple strategy to ensure a reliability-driven error recovery. When an error is detected, we update the execution of the bioassay as follows:

- The operation with error is re-executed.
- The electrodes that may lead to errors will not be used in other operations. Note that for each operation, the on-chip resources occupied by it are all recorded by the control software. Thus depending on the error droplet, it is easy to backtrace to the region where error occurs. We consider all electrodes in this region as the possible locations for defects. These electrodes will therefore be discarded.

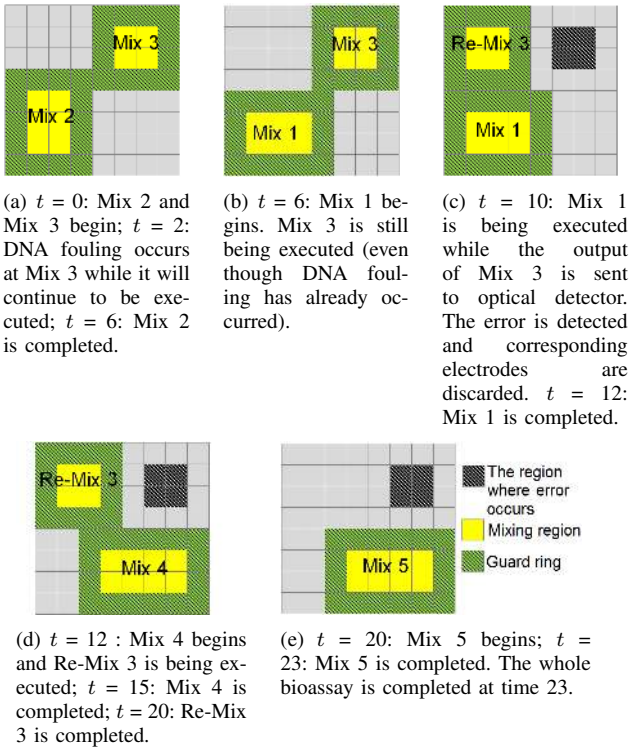
C. Comparison Between Two Sensing Schemes

The two sensing schemes introduced in Section II have differences in the context of fault diagnosis, error recovery and dynamic re-synthesis.

The diagnosis of an electrode with trapped charge can be used to illustrate the difference between these two systems. Suppose a splitting operation with unbalanced droplets occurs, as shown in Figure 9(b). In the CCD camera-based sensing system, Electrode 1 can easily be identified as the electrode with residual charges because the droplet volume is smaller than normal volume.

On the other hand, for the optical detector-based sensing system, the outputs of splitter, which consists of Electrodes 1, 2 and 3 shown in Figure 9(b) will not be used any more. In contrast to the diagnosis result of CCD camera-based sensing system, Electrode 2 and 3 can no longer be used, leading to wastage of on-chip resources. Here we use the bioassay shown in Figure 1 to illustrate the differences of these two sensing schemes.

Suppose the droplets for dispensing operations 1 to 6 in Figure 1(a) are generated from different dispensing ports. Table I shows the synthesis results for all mixing operations



(a) t = 0: Mix 2 and Mix 3 begin; t = 2: DNA fouling occurs at Mix 3 while it will continue to be executed; t = 6: Mix 2 is completed.

(b) t = 6: Mix 1 begins. Mix 3 is still being executed (even though DNA fouling has already occurred).

(c) t = 10: Mix 1 is being executed while the output of Mix 3 is sent to optical detector. The error is detected and corresponding electrodes are discarded. t = 12: Mix 1 is completed.

(d) t = 12 : Mix 4 begins and Re-Mix 3 is being executed; t = 15: Mix 4 is completed; t = 20: Re-Mix 3 is completed.

(e) t = 20: Mix 5 begins; t = 23: Mix 5 is completed. The whole bioassay is completed at time 23.

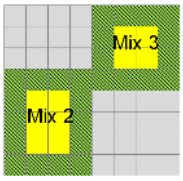
Fig. 11. Synthesis results for the bioassay when we use the optical detector-based sensing system.

of the bioassay shown in Figure 1(a). The module placement result corresponding to synthesis result in Table I is shown in Figure 10.

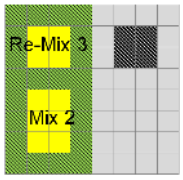
It is important to note that, in Table I, "resource" refers to part of the electrode array occupied by the mixing operation. The location of a mixer is expressed in terms of the location of the electrode at the upper left corner of the mixer. For example, the upper left corner of the mixing module M_1 is in the sixth row and second column; it includes an electrode array with two rows and three columns. Thus the mixer is described as a 3×2 mixer at the location (2, 6).

Suppose that the DNA-fouling phenomenon occurs in the operation Mix 3 shown in Table I after the operation has been underway for three seconds. For the optical detector-based sensing system, the output of Mix 3 is checked only after Mix 3 has been completed. Thus, the error recovery process is triggered at time instant t = 10. For the CCD camera-based system, the error-recovery process will be triggered immediately after DNA fouling occurs at time instant t = 3. The synthesis results for these two cases are shown in Figure 11 and Figure 12, respectively. We find that in the CCD camera-based sensing system, recovery can be triggered immediately after an error occurs. On the other hand, in a detector-based sensing system, recovery can only be triggered at the end of the erroneous operation.

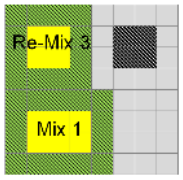
Even though the CCD camera-based sensing system can precisely locate the position of electrodes with defects and trigger recovery immediately after an error occurs, light from the camera may influence some biochemical substances, e.g., fluorescent markers in the droplet [25]. Thus in order to mon-



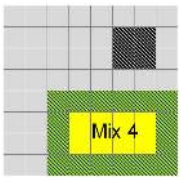
(a) t = 0: Mix 2 and Mix 3 begin; t = 2: DNA fouling occurs, Mix 3 stops and will be re-executed; corresponding electrodes are discarded.



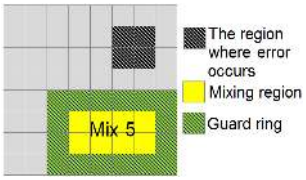
(b) t = 2: recovery operation Re-Mix 3 begins; t = 6: Mix 2 is completed.



(c) t = 6: Mix 1 begins and Re-Mix 3 is still being executed; t = 12: both Mix 1 and Re-Mix 3 are completed.



(d) t = 12: Mix 4 begins; t = 15: Mix 4 is completed.



(e) t = 15: Mix 5 begins; t = 18: Mix 5 is completed. The whole bioassay is completed at time 18.

Fig. 12. Synthesis results for the bioassay when we use the CCD camera-based sensing system.

TABLE II
COMPARISON BETWEEN CCD CAMERA-BASED SENSING SCHEME AND DETECTOR-BASED SENSING SCHEME.

	CCD camera-based scheme	Detector-based scheme
Accuracy of locating electrodes with defect	High	Low
Response time	Error recovery can be triggered immediately when an error occurred	Error recovery can only be triggered at the end of the faulty operation
Application for photosensitive samples/reagents	Cannot be used.	Can be used.

itor experiments that include photosensitive samples/reagents, we need to choose the detector-based sensing scheme.

IV. Error Recovery and Dynamic Re-synthesis

With the availability of hardware that can send feedback to the control software, it is now necessary to design physical-aware software that can analyze sensor data and dynamically adapt to it. Adaptations include updates for the schedule of fluid-handling operations, resource binding, module placement, and droplet routing pathways.

The task of the control software includes two phases: the first phase is off-line data preparation before the execution of bioassay and the second phase is on-line monitoring for the fluid-handling operations as well as dynamic re-synthesis of the bioassay. Details are presented below.

A. Off-line Data Preparation before Bioassay Execution

The first step in data preparation is to convert the sequencing graph of the bioassay to a directed acyclic graph (DAG) and store it in memory for use by the control software. In this DAG, the vertices represent microfluidic handling operations and the edges represent precedence relations between operations. The predecessors and successors of any operation can be determined by performing depth-first search on the graph [26].

The second step in data preparation is to assign error thresholds for each operation. These thresholds are determined by the requirement of precision for the bioassay and they are stored as a table in memory for use by the control software. During bioassay execution, if an optical detection result is outside the range of pre-assigned threshold values, we conclude that an error has occurred at the corresponding operation.

The last step in data preparation is the initial synthesis step for the bioassay. In this procedure, we map the sequencing graph of the bioassay and on-chip resources to the scheduling, resource binding, module placement, and droplet routing results for each operation.

For a sequencing graph consisting of n operations, the synthesis result can be written as the following set:

$$\mathcal{S} = \{M^*_{opt_1}, M^*_{opt_2}, ...M^*_{opt_n}\}$$

where $M^*_{opt_i}$, $1 \leq i \leq n$, is the synthesis output for the ith operation opt_i. The element $M^*_{opt_i}$ can be viewed as an ordered 6-tuple:

$$M^*_{opt_i} = < ts(opt_i), te(opt_i), x(opt_i), y(opt_i), col(opt_i), row(opt_i) >$$

where ts and te are the start time and end time of the operation, respectively; x and y are the x-coordinate and y-coordinate for the module that implements the operation; col and row are the number of columns and rows occupied by the operation in the array.

For an arbitrary operation opt_i, the order of elements in the tuple $M^*_{opt_i}$ is defined. Thus we can use $M^*_{opt_i}(\widetilde{j})$ to represent the $\widetilde{j}$th element in the tuple $M^*_{opt_i}$. For example, the start time of ith operation can be written as $M^*_{opt_i}(1)$, the x-coordinate of ith operation can be written as $M^*_{opt_i}(3)$, and the number of columns occupied by opt_i can be written as $M^*_{opt_i}(5)$.

For simplicity, we write the set of all operations in the bioassay as $\mathcal{P}$; and we use $\mathcal{C}$ to refer to the set of constraints that $\mathcal{S}$ must satisfy, which include:

1) For any pair of operations opt_w and opt_v, if the two open intervals $(M^*_{opt_w}(1), M^*_{opt_w}(2))$ and $(M^*_{opt_v}(1), M^*_{opt_v}(2))$ overlap, i.e.,

$$(M^*_{opt_w}(1), M^*_{opt_w}(2)) \cap (M^*_{opt_v}(1), M^*_{opt_v}(2)) \neq \emptyset,$$

which implies that operations opt_w and opt_v are implemented concurrently. It is important to note that multiple operations cannot share on-chip resources (including electrodes, dispensing ports and detectors) at the same time. Thus opt_w and opt_v must satisfy following constraint:

$$(M^*_{opt_w}(3), M^*_{opt_w}(3) + M^*_{opt_w}(5)) \cap (M^*_{opt_v}(3), M^*_{opt_v}(3) + M^*_{opt_v}(5)) = \emptyset, \vee$$
$$(M^*_{opt_w}(4), M^*_{opt_w}(4) + M^*_{opt_w}(6)) \cap (M^*_{opt_v}(4), M^*_{opt_v}(4) + M^*_{opt_v}(6)) = \emptyset,$$

i.e., their corresponding modules cannot overlap with each other.

2) For any pair of operations opt_w and opt_v, if opt_w is the predecessor of opt_v, then opt_w must be completed earlier than the start time of opt_v, i.e. $M^*_{opt_v}(1) \geq M^*_{opt_w}(2)$.

The completion time of the bioassay can be written as:

$$C_p = \underset{opt_i \in \mathcal{P}}{\text{Max}} \{M^*_{opt_i}(2)\}$$

Thus the synthesis of the biochip can be viewed as an optimization problem. The inputs are the set of operations $\mathcal{P}$ and the set of constraints $\mathcal{C}$. The target is:

$$\text{minimize: } \underset{opt_i \in \mathcal{P}}{\text{Max}} \{M^*_{opt_i}(2)\}$$

Previously published CAD methods for digital microfluidic biochip have proposed several algorithms to solve this optimization problem. For example, the PRSA-based synthesis algorithms can be used to quickly derive optimized synthesis results [27].

After the optimized synthesis results are derived, the off-line data preparation step is completed. The bioassay is next executed according to the initial synthesis result, and the next step is the on-line monitoring of droplets.

B. On-line Monitoring of Droplets and Re-synthesis of the Bioassay

During the execution of the bioassay, the control software must implement the following steps.

Step 1: Error Identification

The error identification procedures for the optical detector-based sensing system and CCD camera-based sensing system are different. For the detector-based sensing system, the outputs of each operation are sent to an on-chip detector. After each optical detector operation, the software compares the detection result with a pre-assigned error threshold. If the optical detection result fails to meet the requirement of the experiment, we conclude that an error has occurred. The detection of the error will trigger the error recovery procedure, i.e., the software will dynamically adjust the synthesis results to re-execute the operation in which an error occurred and bypass the electrodes at which the error occurred.

For the CCD camera-based sensing system, the software can carry out a real-time monitoring of all droplets on the biochip. The colors and diameters of the droplets are detected simultaneously by the CCD camera and evaluated for comparisions. Thus, error recovery is triggered as soon as an error occurs.

According to the detection results for droplets, both optical detector-based and CCD camera-based systems can locate the on-chip resources with defects. Suppose the output droplet of operation opt_o fails to meet the requirement, then we can conclude the defect may exist in the resources which are assigned to opt_o. In order to ensure the reliability of the subsequent operations, this region will be bypassed.

Step 2: Update of Sequencing Graph

When an error occurs, the control software determines the required recovery operations. According to the above discussion, if an error occurs during a reversible operation, the recovery process is simple. For non-reversible operations, the control software must search the preceding operations until it finds an operation that can provide backup droplets to feed the inputs of the recovery subroutine. As mentioned above, when an error occurs during the implementation of operations, the software will adjust the sequencing of the bioassay according to the category of operation. The pseudocode for the adjustment of the sequencing graph is shown in Figure 13. The definition of error recovery graph G_{Re_i} for operation opt_i can be found in Part A of Section III. The update of the sequencing graph for errors that occur during the operations of Categories I, II, and III are shown in Figure 14. The categorization of operations can be found in Part A of Section III.

1: Derive the graph $G_{original}$ by deleting edges between the erroneous operation and its immediate predecessors in original sequencing graph;
2: Derive the error recovery graph G_{Re_i} for opt_i;
3: Copy G_{Re_i} and label the nodes with different names;
4: Derive the union graph for G_{Re_i} and $G_{original}$;

Fig. 13. Pseudocode for adjustment of the sequencing graph.

It is important to note that, for some operations, the recovery subroutines may change depending on the error. For example, operation 7 in Figure 7(a) generates two droplets; one of them is used in the subsequent reaction and the other is stored on chip as the "backup droplet". If a single error occurs at operation 14, the biochip will re-execute operations 8, 9, 10 and 12. However, if an error occurs at a predecessor of operation 14, the recovery subroutine for operation 14 will be different. For example, when an error occurs at operation 9, the backup droplet of operation 7 will be used as the input for error recovery. If another error occurs afterwards at operation 14, there is no more backup droplet available at the output of operation 7. Thus the recovery subroutine of operation 14 has to be expanded and it will now include operation 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, and 12. Therefore the recovery steps are completely different from the case when an error occurs at operation 14.

After the recovery subroutine of an operation is determined, the control software will update the sequencing graph and the corresponding DAG, and then the dynamic re-synthesis step will be implemented.

Step 3: Dynamic Re-synthesis

In the cyberphysical system envisioned here, when an error is detected at a checkpoint, it will trigger the generation of a new mapping of the remaining steps (including proper handling of intermediate results) of the bioassay. This process is referred to here as *re-synthesis*, on the basis of the initial design obtained from the *a priori* synthesis step. The requirements for the process of re-synthesis are:

- The interruption of other operations should be avoided. Consider the following example. For the bioassay with synthesis results shown in Table I, suppose an error recovery process is triggered by an error in operation Mix 3 at time instance 10. When the error recovery process is triggered, operation Mix 1 is being implemented. In order

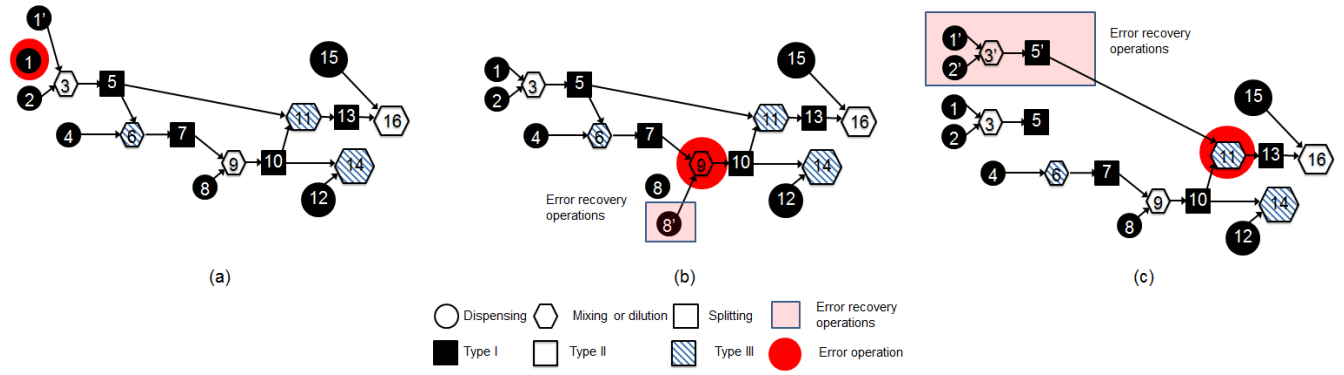


Fig. 14. Update of the sequencing graph corresponding to error operations of Categories (a) I; (b) II; and (c) III.

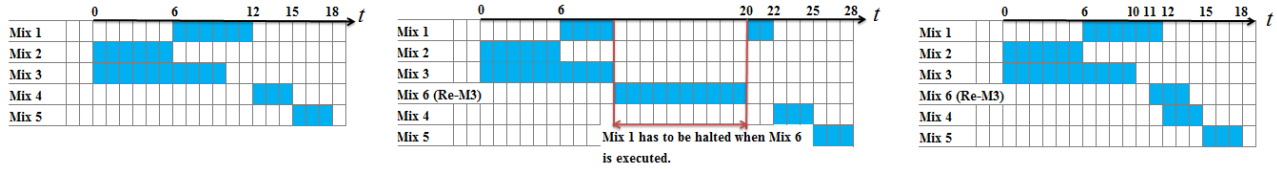


Fig. 15. (a) Scheduling result when no error occurs; (b) scheduling when an error occurs in Mix 3. Mix 1 is halted when error operations are executed; (c) scheduling when an error occurs in Mix 3. Here dynamic synthesis strategy is applied at time 10 and error recovery operations begin at time 11.

to avoid the interruption of Mix 1, in the new synthesis results, the schedule and resource assignment results for Mix 1 should be the same as in the inial synthesis results.

- The electrodes at which an error has been deemed to have occurred should be bypassed in the new synthesis results.
- The completion time of the bioassay should be optimized.

To satisfy these requirements, we propose two re-synthesis strategies for dynamic re-synthesis. The first strategy is based on a local greedy algorithm, and the second is a PRSA-based, global optimization algorithm [3].

For the greedy algorithm, the first step is to determine all operations that must be adjusted in the re-synthesis result. These operations include the operations in the error recovery graph, the erroneous operation, and the set of subsequent operations that will be implemented on electrodes with defects in the initial synthesis result. Other operations will be executed based on the initial synthesis result.

Since the synthesis results for part of the operations are fixed, dynamic re-synthesis on the microfluidic array can be modeled as the *module placement with obstacles* problem. Here the operations that are implemented based on the initial synthesis result are fixed *a priori* as the "obstacles" while the other operations that are necessary for recovery are derived through re-synthesis and placed in the remaining available chip area in a greedy fashion. The detailed steps are described below.

First the control software places all operations that need to be re-scheduled in a priority queue based on topological sort. These operations include error recovery operations and all successors of the erroneous operation. Then the software assigns a priority for each operation in the queue. The "deepest" operation in the subroutine (i.e., the operation at the bottom of the list generated by topological sort) is assigned the lowest priority while the "shallowest" (at the top of the list produced by topological sort) operation is assigned the highest priority in the queue.

Next the control software needs to allocate on-chip resources to these operations. Here our on-chip resource set R changes with time t. The control software will search for available resources at the current time for the operation with the highest priority. For example, if the operation with the highest priority is a mixing operation, then the system will search for an available $m \times n$ cell array that is not occupied from current time t to t+$\triangle$t. Here $\triangle$t is the time needed for the operation in the $m \times n$ cell array. If suitable idle resources are available, resource binding will be successful and the start time of the operation will be deemed to be the current time. Otherwise, the operation has to be delayed until there are available resources. If multiple resources are available at the same time, the control software will randomly choose one and bind it to the corresponding operation. After resource binding and start/stop time of the operation are determined, the operation will be removed from the priority queue.

Note that the above steps can also be used to generate re-synthesis results, when multiple errors are detected at the same time. In this situation, multiple recovery processes are triggered at the same time and the control software generates a priority queue for each recovery process. After these priority queues merged, the control software assigns a priority for each element based on topological sort. Finally, the control software determines new synthesis results for every operation in the merged priority queue.

For a microfluidic biochip with an $M \times N$ electrode array, D detectors and P dispensing ports, the computational complexity of searching for available resources (i.e. "the maximum empty rectangle") in this re-synthesis algorithm is O($MN + D + P$). Since we can view the number of detectors and dispensing ports as being constant and we are

1: Localize the fault operation according to feedback at checkpoints;
2: Determine the operations which need to be adjusted and store them into a priority queue Q;
3: Delete all initial synthesis results for operations in Q;
4: **while** $Q \neq \emptyset$ **do**
5: Seach avaibale resource for operation q_0 which has the highest priority in Q;
6: Remove q_0 from Q;
7: **end while**

Fig. 16. Pseudocode for dynamic re-synthesis of the bioassay.

interested in algorithm scalability for large arrays, the worst-case complexity is O(MN). This is because the software will exhaustively search each electrode/detector/dispensing port in the array and check whether it is available. The computational complexity for other parts of the algorithm are all O(1). Hence the overall computational complexity of the re-synthesis algorithm is O(MN). The pseudocode for the re-synthesis procedure is shown in Figure 16.

An example of re-synthesis is shown in Figure 15. Figure 15(a) shows the schedule corresponding to the sequencing graphs in Figure 1(a). Figure 15(b) and Figure 15(c) both show the schedules corresponding to the sequencing graph in Figure 1(b). For the sake of clarity, we only show the schedule for mixing operations. Here Figure 15(b) is the schedule obtained using the error-recovery algorithm of From Fig. 15(b), we can see that mixing operation Mix 1 is halted for 10 time slots when error recovery operations are executed. The completion time of the bioassay shown in Fig. 15(a) is increased from 18 time slots to 28 time 18 time slots to 28 time slots, which can be unacceptable for many applications. The dynamic scheduling result corresponding to Fig. 1(b) is shown in Fig. 15(c). When the error is detected at the output of Mix 3 at time 10, the Mix 3 at time 10, the ongoing operation Mix 1 is executed based on the initial synthesis result. We assume that the computing time to generate the new synthesis result is 1 time slot. In practice, computation time is at least an order of magnitude less than the fluidic operation time. Then at time 11, the control software will generate new synthesis result based on the updated Fig. 1(b). As shown in Fig. 15(c), Mix 1 is completed at time 12 without being interrupted and the being interrupted and the experiment is finished at time 18. Thus the bioassay is executed "seamlessly" without any time penalty or interruption of other operations.

The re-synthesis problem can also be solved using the PRSA-based global optimization method from [3]. The inputs and constraints of the re-synthesis problem are different from the initial synthesis problem introduced in Section III.A. Suppose the set of operation for the re-synthesis problem is $\mathcal{P}'$ and the set of constraints is $\mathcal{C}'$. We can derive $\mathcal{P}'$ and $\mathcal{C}'$ based on $\mathcal{P}$ and $\mathcal{C}$ introduced in Section IV.

We first define a operator $\mathcal{T}$ on the set $\mathcal{P}$. $\mathcal{T}$ is a mapping from the set of all operations to the set of operations that have already started at time instant t.

$$\mathcal{T}(t) : \mathcal{P} \rightarrow \mathcal{P}(t) = \{opt_i | M^*_{opt_i}(1) \leq t\}$$

When an error is detected at time instant t in operation opt_i, the set of operations that need to be re-synthesized can be written as $\mathcal{P}' = \mathcal{P} \cup \mathcal{R}_i \cup \widetilde{\mathcal{O}} - \mathcal{P}(t)$. Here $\mathcal{R}_i$ is the set of recovery operations corresponding to erroneous operation opt_i, and $\widetilde{\mathcal{O}}$ is the set of subsequently operations which will be implemented on electrodes with defect in the initial synthesis result. The method for determining the operations in $\mathcal{R}_i$ is introduced in Section III.A. Then according to the module placement information included in initial synthesis result, and locations of electrodes with defects, operations in $\widetilde{\mathcal{O}}$ can be determined.

We write the new synthesis results for $opt_i \in \mathcal{P}'$ as M'_{opt_i}. In addition to the set of constraints $\mathcal{C}$, the new synthesis must satisfy the constraint that the region where an error has been deemed to have occurred cannot be used any more.

The optimization problem for re-synthesis process can be written as:

$$\text{minimize: } \underset{opt_i \in \mathcal{P}'}{\text{Max}} \{M'_{opt_i}(2)\}$$

The above optimization problem can be solved by using the PRSA-based synthesis procedure introduced in [3]. Using this method, we can derive globally-optimized synthesis results with short assay completion time, but the CPU time is high of the order of 20 minute for a typical bioassay [13]. Thus, this method is not suitable for on-line computation of re-synthesis results.

V. SIMULATION RESULTS

In this section, we evaluate the re-synthesis approach for error recovery on representative bioassays that are especially prone to fluidic errors. We compare the completion time for the two sensing schemes; and the re-synthesis results derived by the greedy algorithm and the PRSA-based global optimization algorithm.

A. Preparation of Plasmid DNA

First, we simulate the preparation of plasmid DNA by alkalinelysis with SDS-minipreparation [31]. During sample preparation, a mixture of three reagents is required. The three reagents are:

• Reagent R_1: Alkaline lysis Solution I (50 mM Glucose,25 mM Tris-HCl (pH 8.0), 10 mM EDTA (pH 8.0)).

• Reagent R_2: Alkaline lysis Solution II (0.2 N NaOH, 1% SDS (w/v)).

• Reagent R_3: Alkaline lysis Solution III (5 M sodium acetate, glacial acetic acid).

The required concentration of the mixture is 22% of R_1, 44% of R_2, and 34% of R_3, which can be approximated to $\frac{28}{128}$ of R_1, $\frac{56}{128}$ of R_2, and $\frac{44}{128}$ of R_3. The sequencing graph to get the required concentration by mixing R_1, R_2, and R_3 is shown in [17]. This bioassay is mapped to a 10×10 electrode array and all electrodes on the boundary of the array are used as storage cells.

The error-recovery capability of the cyberphysical microfluidic system can be evaluated on the basis of the bioassay completion time when errors are detected. We inject errors randomly into the chip during the execution of the bioassay and compare the completion time of the two sensing schemes. The results are shown in Figure 17. Here the completion time

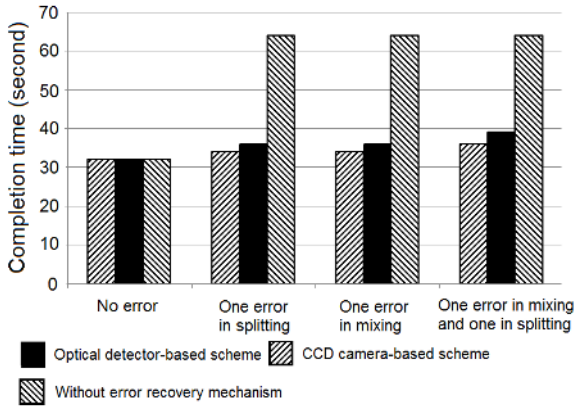


Fig. 17. Completion time for biochip with detector-based and CCD camera-based sensing system, and the biochip without error recovery mechanism when errors are injected in the sample preparation of plasmid DNA.

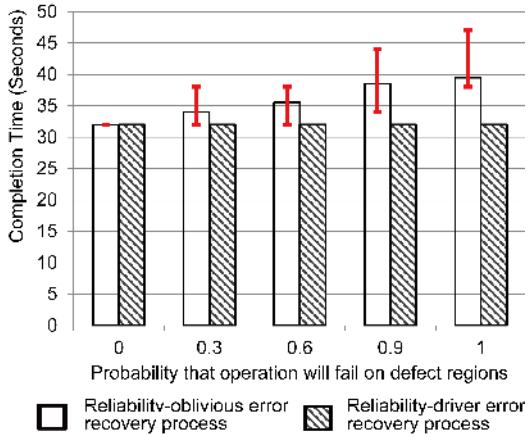


Fig. 18. Comparison for the completion time between reliability-driven and reliability-oblivious error recovery [17] when a 1×4 subarray is defective in the sample preparation of plasmid DNA. The error bars show the maximum and minimum completion time for reliability-oblivious error recovery in simulation.

is derived from the greedy algorithm introduced in Section IV, and the results are the average of the values derived from repeating the experiments 10 times.

From Figure 17, we can see that the completion time for the CCD camera-based sensing scheme is slightly less than the completion time for the optical detector-based scheme. This is because in the CCD camera-based scheme, when an error occurs during an operation, the error recovery procedure is immediately triggered; in the detector-based scheme, the recovery procedure only can be triggered at the end of the erroneous operation. The simulation results for biochip without error recovery mechanism are also shown in Figure 17. For this case (no error recovery), if an error occurs during the bioassay, the final outcome of the entire experiment will be incorrect. As a result, the biochip has to be discarded, and the experiment must be repeated on a new biochip in order to correct the error. If we assume that re-execution of the experiment will be successful, the bioassay completion time will be twice as the completion time in the fault-free case. Based on these results, we note that error recovery can reduce the bioassay completion time, and the consumption of biochemical reagents/samples can be reduced.

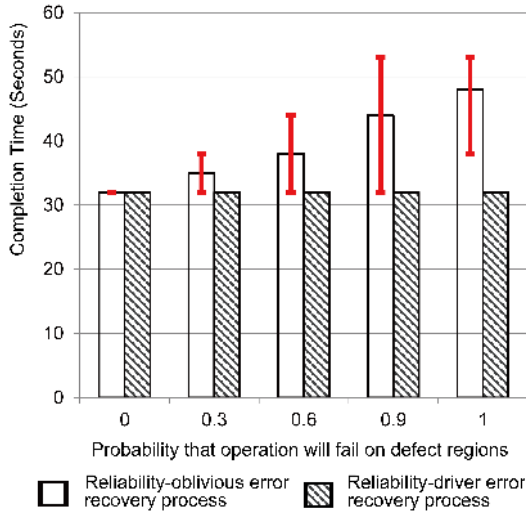
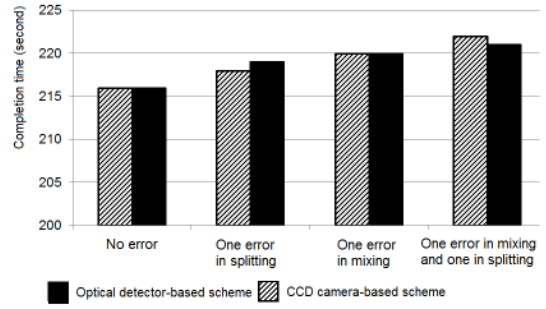


Fig. 19. Comparison between the completion time of reliability-driven and reliability-oblivious error recovery when a 2×4 subarray is defective in the sample preparation of plasmid DNA. The error bars show the maximum and minimum completion time for reliability-oblivious error recovery in simulation.



(a)

(b)

Fig. 20. (a) Completion time for biochips with detector-based and CCD camera-based sensing systems when errors are injected in the sample preparation of interpolating mixing; (b) Completion time for biochips with detector-based and without error recovery mechanism when errors are injected in the sample preparation of interpolating mixing.

In reliability-driven error recovery, the electrodes where an error is deemed occurred will not be used in other operations. On the contrary, for the reliability-oblivious error recovery process in [17], when an error occurs during execution, the region where error occurs will continue to be used in subsequent operations. As discussed in Section III.B, these electrodes with defects may further lead to more errors.

To compare between the completion time for reliability-

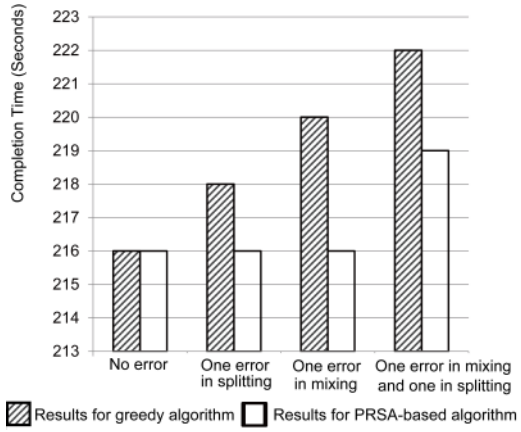


Fig. 21. Completion time for the bioassay of interpolating mixing (derived from two re-synthesis algorithms when multiple errors are injected).

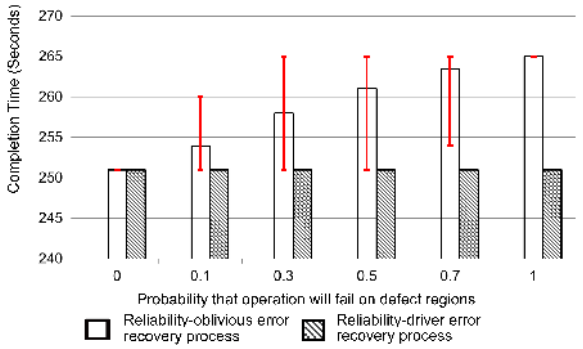


Fig. 22. Comparison between the completion time of reliability-driven and reliability-oblivious error recovery [17] when a 1×4 defect array is injected in exponential dilution. The error bars show the maximum and minimum completion time for reliability-oblivious error recovery in the simulation.

oblivious and reliability-driven error recovery procedures, the following simulation was set up. In the reliability-oblivious error recovery, we randomly select one operation opt_{fe} as the first instance of error in the execution of bioassay. The electrodes that are used to implement opt_{fe} are referred to "electrodes with defects". When another operation is implemented again on these electrodes with defects, we assume that there exists a probability P_{fail} that this operation will also fail. For a fixed value of P_{fail}, we simulate reliability-oblivious error recovery 15 times, and determine average completion time.

Figure 18 compares the completion time of reliability-driven error recovery and average completion time of reliability-oblivious error recovery for different values of P_{fail}. Here the randomly selected opt_{ef} is a mixing operation implemented on a 1×4 electrode array. As expected, Figure 18 shows that the reliability-driven error recovery leads to shorter assay completion time in the presence of defects.

Next we randomly select another operation as opt_{fe} and run the simulation again. The electrodes that implement opt_{fe} now constitute of a 2×4 electrode array. The simulation results are shown in Figure 19. We find that as expected, the average completion time for reliability-oblivious error recovery is higher when more electrodes are likely to be defective. On the other hand, the completion time of reliability-driven error recovery does not depend on the type of defect on the chip and keeps the minimum completion time.

B. Protein Assays: Interpolating Mixing and Exponential Dilution

Next we evaluate re-synthesis and error recovery for two real-life protein assays. These assays lead to the dilution of a protein sample by using two methods, namely interpolating mixing and exponential dilution. The protocols and corresponding sequencing graphs for these two bioassays are described in [13].

When errors are injected in the sample preparation of interpolating mixing, the completion time of biochips with detector-based and CCD camera-based sensing systems are shown in Figure 20(a); and the completion time of biochips with detector-based sensing systems and without error recovery mechanism are shown in Figure 20(b). The bioassay is mapped to a 10×10 electrode array.

Figure 21 reports the completion time when multiple errors are inserted during the interpolating mixing protocol. Note that the completion time defined here only includes time spent on fluid-handling operations, and excludes the CPU time spent on resynthesis. From Figure 21, we see that the completion time achieved by the PRSA-based algorithm and greedy algorithm are almost the same, but the CPU times for these two algorithms are different. The experiment was performed on a 2.6-GHz, Intel i5 processor with 6 GB of memory. Both re-synthesis algorithm are implemented on the basis of the same initial synthesis result. The CPU time needed was around 33 minutes for computing the re-synthesis results using PRSA, which was 10 times more than the bioassay completion time; while the CPU time was less than 5 seconds for the greedy algorithm, which was only 2.5% of the bioassay completion time. The bioassay completion time derived by greedy algorithm is only slightly higher for the PRSA. Nevertheless, the greedy algorithm is more suitable for on-line re-synthesis because of low CPU time.

While the PRSA-based approach is less attractive for real-time decision making, it provides a useful calibration point for the greedy algorithm and shows that the latter's effective for timely bioassay completion. Moreover, the PRSA-based method can serve as the basis for future error recovery methods based on precomputation and preloading of recovery schedules.

For the exponential dilution protocol introduced in [13], we evaluate the completion time for the reliability-driven and reliability-oblivious error recovery methods in Figure 22. First we randomly select one operation opt_{fe} as the first instance of error in the execution of bioassay, where opt_{fe} is a dilution operation implemented on a 1×4 electrode subarray. Then for subsequent operations that are implemented on this electrode array with defects, we set P_{fail} as the probability that the operation will fail again. Then corresponding to each value of P_{fail}, we run the simulations 15 times, and derive the average completion time for reliability-oblivious error recovery. In contrast, the defective electrodes are bypassed in reliability-driven error recovery. Thus the completion time of reliability-driven error recovery is independent of P_{fail}. From

the results shown in Figure 22, we find that reliability-driven error recovery reduces the bioassay completion time. At the same time, we avoid the problem that any given set of defective electrodes can lead to replicated errors, thus the number of errors in the bioassay is reduced. Hence less reagents/samples are consumed, leading to lower cost and higher reliability for the experiment.

VI. Conclusions

We have shown how recent advances in the integration of optical sensors in a digital microfluidics biochip can be used to make biochips error-resilient. We have presented a cyberphysical approach for "physical-aware" system reconfiguration that uses sensor data at intermediate checkpoints to dynamically reconfigure the biochip. Real-time experiment monitory techniques based on integrated optical detector and CCD camera have been considered. Two different sensor-driven re-synthesis techniques have been developed to re-compute electrode-actuation sequences, thereby deriving new schedules, module placements, and droplet routing pathways, with minimum impact on the time-to-response. These two methods have been compared in terms of bioassay completion time and CPU time needed for re-synthesis. The coordination between the physical-aware control software and the microfluidic biochip allows sensor data at intermediate checkpoints to be used as feedback to make decisions about completed operations, and dynamically reconfigure the biochip and optimize electrode actuation sequences for subsequent operations. The proposed approach has been evaluated and its effectiveness demonstrated for three representative protein bioassays.

References

[1] H. Becker, "Microfluidics: a technology coming of age", *Medical Device Technology*, vol. 19, 2008.

[2] R. B. Fair, "Digital microfluidics: Is a true lab-on-a-chip possible?", *Microfluidics and Nanofluidics*, vol. 3, pp. 245-281, 2007.

[3] F. Su and K. Chakrabarty, "High-level synthesis of digital microfluidic biochips", *ACM J. Emerging Tech. in Comp. Sys.*, vol. 3, January 2008.

[4] E. Maftei et. al., "Routing-based synthesis of digital microfluidic biochips", *Proceedings of the 2010 International conference on Compilers, Architectures and Synthesis for Embedded Systems*, pp. 41-50, 2010.

[5] T.-W. Huang, C.-H. Lin, and T.-Y. Ho, "A contamination aware droplet routing algorithm for the synthesis of digital microfluidic biochips", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 11, pp. 1682-1695, November 2010.

[6] T.-W. Huang and T.-Y. Ho, "A two-stage ILP-based droplet routing algorithm for pin-constrained digital microfluidic biochips", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol 30, no. 2, pp. 215-228, February 2011

[7] M. Iyengar and M. McGuire, "Imprecise and qualitative probability in systems biology", *International Conference on Systems Biology*, 2007.

[8] O. Levenspiel, "Chemical reaction engineering", *New York: Wiley*, 1999.

[9] J. K. Park, S. J. Lee, and K. H. Kang, "Fast and reliable droplet transport on single-plate electrowetting on dielectrics using nonfloating switching method, *J. Biomicrofluidics*, no. 4, 2010

[10] J. Verheijen et. al., "Reversible electrowetting and trapping of charge: model and experiments, *ACS J. Langmuir*, No. 15, pp. 66166620, 1999.

[11] E. Welch et. al., "Picoliter DNA sequencing chemistry on an electrowetting-based digital microfluidic platform", *Biotech. J.*, vol. 6, pp. 165-176, 2011.

[12] C. A. Mein et. al., "Evaluation of single nucleotide polymorphism typing with invader on PCR amplicons and its automation", *Genome Res.*, vol. 10, pp. 330-343, 2000.

[13] Y. Zhao, T. Xu, and K. Chakrabarty, "Integrated control-path design and error recovery in digital microfluidic lab-on-chip", *ACM J. Emerging Tech. in Comp. Sys.*, vol. 3, no. 11, August 2010.

[14] R. Evans et. al., "Optical detection heterogeneously integrated with a coplanar digital microfluidic lab-on-a-chip platform", *Proc. IEEE Sensors Conf.*, pp. 423-426, Oct. 2007.

[15] N. M. Jokerst et. al., "Progress in chip-scale photonic sensing", *IEEE Trans. Biomedical Circuits and Sys.*, vol. 3, pp. 202-211, 2009.

[16] W. Bialek et. al., "Protein dynamics and reaction rates: mode-specific chemistry in large molecules?", *Proceedings of the National Academy of Sciences of the United States of America*, vol. 85, pp. 5908-5912, 1988.

[17] Y. Luo, K. Chakrabarty, and T.-Y. Ho, "A cyberphysical synthesis approach for error recovery in digital microfluidic biochips", *Proc. DATE*, pp. 1239-1244, 2012.

[18] R. Fair, A. Khlystov, V. Srinivasan, V. Pamula, and K. Weaver, "Integrated chemical/biochemical sample collection, pre-concentration, and analysis on a digital micro uidic lab-on-a-chip platform", *Lab-on-a-Chip: Platforms, Devices, and Applications, Conf*, page 5591, 2004.

[19] F. Su, K. Chakrabarty, and R. B. Fair, "Microfluidics-based biochips: technology issues, implementation platforms, and design automation challenges", *IEEE Transactions on Computer-Aided Design of Integrated Circuits & Systems*, vol. 25, pp. 211-223, February 2006.

[20] Y. Shin and J. Lee, "Machine vision for digital microfluidics", *Review of Scientific Instruments*, 2010.

[21] Y. Zhao and K. Chakrabarty, "Digital microfluidic logic gates and their application to built-in self-test of lab-on-chip", *IEEE Transactions on Biomedical Circuits and Systems*, vol. 4, pp. 250-262, August, 2010.

[22] A. Furtado and R. Henry, "Measurement of greenuorescent protein concentration in single cells by image analysis", *Analytical Biochemistry*, pp. 84-92, November 2002.

[23] S. Rodriguez-Cruz, J. Khoury, and J. Parks, "Protein fluorescence measurements within electrospray droplets", *Journal of the American Society for Mass Spectrometry*, pp. 716-725, June 2001.

[24] M. Jebrail, and A. Wheeler, "Let's get digital: digitizing chemical biology with microfluidics", *Current Opinion in Chemical Biology*, vol. 14, pp. 574-581, 2010.

[25] U. Resch-Genger et. al., "Quantum dots versus organic dyes as fluorescent labels", *Nature Methods*, pp. 763-775, 2008

[26] R. Sedgewick, *Algorithms in C: graph algorithms*, Addison-Wesley, Chapter 23, 2001.

[27] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing", Science, vol. 220(4598), pp. 671-680, May 1983.

[28] H. Murata, K. Fujiyoshi, and M. Kaneko, "VLSI/PCB placement with obstacles based on sequence pair", *Proc. ISPD*, pp. 26-31, 1997.

[29] L. Sha et. al., "An analytical algorithm for placement of arbitrarily sized rectangular blocks", *Proc. DAC*, pp. 602-608, 1985.

[30] A. Alon and U. Ascher, "Model and solution strategy for placement of rectangular blocks in the Euclidean plane", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 7, no. 3, pp. 378-386, 1988.

[31] V. Ananthanarayanan and W. Thies, "Biocoder: a programming language for standardizing and automating biology protocols", *Journal of Biological Engineering*, vol. 4, no. 1, 2010.

[32] F. Su and K. Chakrabarty, "Defect tolerance for gracefully-degradable microfluidics-based biochips", *Proc. IEEE VLSI Test Symposium*, pp. 321-326, 2005.