

Chapter 2

Error-Recovery in Cyberphysical Biochips

In this chapter, through exploiting recent advances in the integration of sensing system in a digital microfluidics biochip, we present a “physical-aware” system reconfiguration technique that uses sensor data at intermediate checkpoints to reconfigure the biochip dynamically. A cyberphysical re-synthesis technique is used to recompute electrode-actuation sequences, thereby deriving new results for module placement, droplet routing pathways, and operation schedules, with minimum impact on the time-to-response.

The key contributions of this chapter are as follows:

- A charge-coupled device (CCD)-based sensing system for digital microfluidic biochips (Sect. 2.2).
- An imaging algorithm for the measurement and tracking of droplets based on real-time data from a CCD camera (Sect. 2.2).
- A reliability-driven error-recovery strategy (Sect. 2.3).
- Parallel recombinative simulated annealing (PRSA)-based and greedy algorithms for reliability-driven synthesis (Sect. 2.4).
- Simulation results for three representative bioassays (Sect. 2.5).

2.1 Motivation and Related Prior Work

The ease of reconfigurability and software-based control in digital microfluidics has inspired research on various aspects of automated chip design and chip applications. A number of techniques have been published for architectural-level synthesis [1], module placement, and droplet routing [2–4]. However, these techniques ignore practical realities or domain-specific constraints that arise from attempting to carry out biochemical reactions and microfluidic operations on an electronic chip.

Due to the complex and randomness component interactions that are ubiquitous in biological/chemical processes, predictive modeling and accuracy control become difficult [5, 6].

In addition to manufacturing defects and imperfections, various faults may also arise during bioassay execution. For example, DNA fouling may lead to malfunction of multiple electrodes in the biochip and excessive actuation voltage applied to an electrode may lead to breakdown of electrodes and charge trapping [7–9]. These faults are hard to detect *a priori*, but they occur during bioassays in certain situations [9]. Yet, despite such inherent variability, many biomedical experiments, such as clinical diagnostics and drug development, require fluid-handling operations that are highly accurate and precise. Each step in the protocol of a biochemical experiment has an “acceptance range” for the volume and concentration of droplets. For example, in the preparation of samples of plasmid DNA, the pH of the solution must be less than 8.0 to avoid a significant reduction in the efficiency of the lysozyme [10]. If an unexpected error occurs during the experiment or the requirements of the bioassay protocol are violated, the outcome of the entire experiment are incorrect. When this occurs, all the steps of the experiment must be repeated to correct the error [11, 12]. Repetition of experiments leads to wastage of expensive reagents and hard-to-obtain samples.

The repetitive execution of on-chip laboratory experiments results in the following problems: (i) an increase in the time-to-result for a bioassay, which is detrimental to real-time detection and rapid response; (ii) wastage of samples that are difficult to obtain or prepare, as well as the wastage of expensive reagents.

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 [11], which proposed intermediate stage monitoring and rollback error-recovery for a microfluidic biochip. In [11], sensing system is used to verify the correctness of immediate product droplets at various steps in the on-chip experiment. When an error is detected at a sensor, i.e., the volume or concentration of the droplet is below or above the acceptable calibrated range, the corresponding droplet is discarded. If the outputs of an operation fail to meet the quality requirements that are derived based on sensor calibration, the operation will be re-executed. New product droplets will be generated to replace the unqualified droplet.

Figure 2.1 shows an example of rollback error-recovery. The initial sequencing graph of a bioassay is shown in Fig. 2.1a. Here we assume that the outputs of each dispensing, mixing and splitting operation are evaluated by a sensor. When an error occurs at operation 9, the system will re-execute the corresponding dispensing and mixing operations. Figure 2.1b shows the new sequencing graph for error-recovery, where operations 12, 13, and 14 are added to generate new product droplets. In the absence of “physical-aware” control software, the error-recovery method in [11] suffers from following drawbacks:

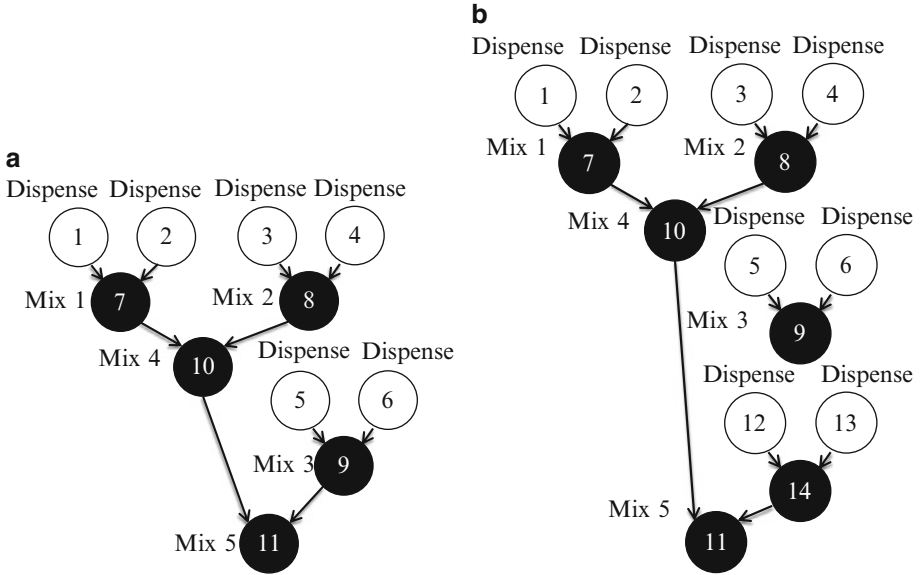


Fig. 2.1 (a) Initial sequencing graph; (b) operations 12, 13, and 14 are added for error-recovery

1. The first drawback is the over-simplification of fault detection and the associated assumptions. It is impractical to use a uniform “expected value” for the calibration of each detection operation. Note that the concentration of intermediate product droplets vary in a dynamic manner at various stages during bioassay execution. Hence the calibration needs to be repeated and carried out dynamically as well.
2. In [11], all recovery operations are carried out in a stand-alone manner. All other ongoing bioassay-related fluidic operations are interrupted when an error is detected. The potential long waiting times introduced by recovery operations will lead to sample degradation and erroneous assay outcomes [14]. Some operations, such as colorimetric enzyme-kinetic reactions, require precise durations as specified by the reaction protocol, and they cannot be elongated without introducing unpredictability in the experiment outcome [15].
3. The error-recovery approach in [11] cannot handle situations when multiple errors occur during a bioassay. For example, [11] 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 [11] 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 [7, 8]. If the use of such electrodes is continued, it is likely that they will introduce more errors [7, 8]. 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 sensor feedback-driven and closed-loop biochip operations under software control. By exploiting recent advances in the integration of sensing system in a digital microfluidics biochip [16], we present a “physical-aware” system reconfiguration technique that uses sensor data at intermediate checkpoints to dynamically reconfigure the biochip [17].

2.2 Overview of Cyberphysical Biochips

In this section, we introduce the cyberphysical system on microfluidic biochips and as well as each component of the system. 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 will appropriately reconfigure the microfluidic biochip. In this way, the various steps in the bioassay are executed based upon real-time sensing of intermediate results.

Figure 2.2 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. In this way, the error will be corrected.

2.2.1 Sensing Systems

Two sensing schemes can potentially be used in the cyberphysical system on digital microfluidic biochips.

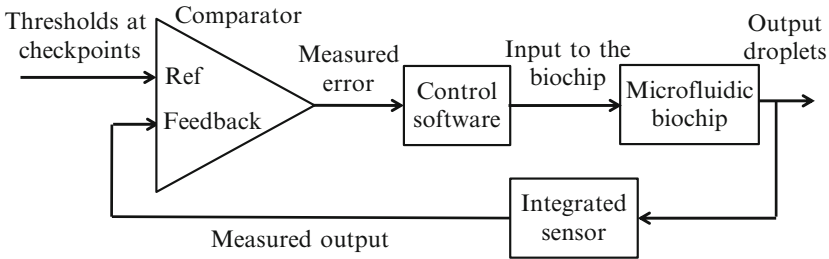


Fig. 2.2 The schematic workflow of the cyberphysical digital microfluidic system

The first sensing scheme is CCD camera-based. As described in Chap. 1, CCD cameras can be used in experiments to view the top sides of droplets simultaneously.

Based on images captured by the CCD camera, droplets can be automatically located by the control software. The procedure for automatical search of the 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 measures the similarity between the template and the “cropped image”. The correlation factor is calculated on a pixel-by-pixel basis, and this process is shown in Fig. 2.3a.

In the control software, all images are stored in grayscale form. These grayscale images can be encoded as matrices or vectors. Suppose the template image is represented in a 1-D array: $\mathbf{x} = (x_1, x_2, \dots, x_N)$. Here x_i represents the gray 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 $\mathbf{y} = (y_1, y_2, \dots, y_N)$. 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 correlation factor with larger absolute value represents a stronger relationship between two images.

After deriving the correlation factors for all possible locations in the image of the complete biochip, we obtain the correlation map between the template and the original input image. Suppose there are κ droplets on the biochip. The locations of droplets can be determined by searching for the largest κ correlation factors in the correlation map. An example is shown in Fig. 2.3b, c [18].

Figure 2.3b shows the original input image of the whole chip and the pattern image, and Fig. 2.3c 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. According to the image, the sizes and colors of droplets can be further analyzed. 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 techniques to check whether the droplets have been moved to the expected positions. This procedure is implemented using the following steps:

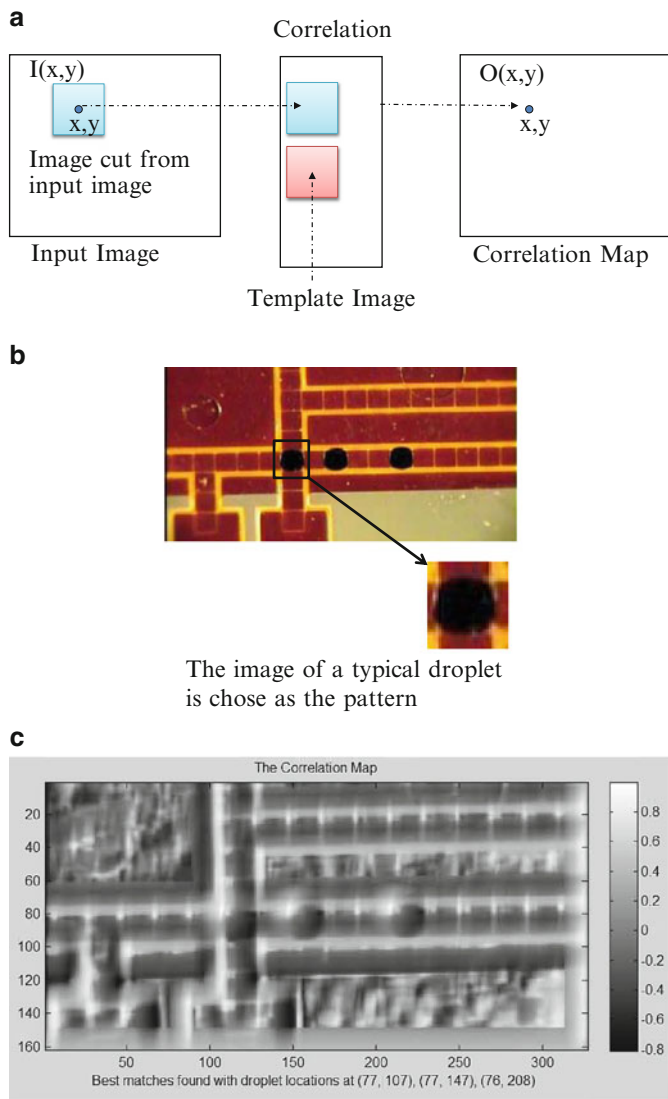


Fig. 2.3 (a) The matching process moves the template image to all possible positions in a larger source image and computes a numerical index that indicates how well the template matches the sub-image in that position; (b) the image of the whole biochip [18] and the pattern we selected; (c) the correlation 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

First, some calibrations are made before the experiment. We choose a large number of sub-images with (or without) droplets, and calculate their correlation with

the template. Based on the calculation results, we search 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 identification of the precise locations of the errors, and (ii) the detection of errors immediately after they occur. One 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 [15, 16]. By examining the concentration of the product in the droplets through fluorescence, the quality of an intermediate product in a digital microfluidic biochip can be determined [15, 16].

When a fluorophore tag is attached to a droplet, different product concentrations lead to the emission of light with different spectrum (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 [16]. In recent work, integrated photodetectors have been introduced on the microfluidic array [15, 16]. 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 can be calculated according to the output voltage of the photodiode. 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. A coplanar digital microfluidic chip with the integrated InGaAs photodetector is shown in Fig. 2.4 [15].

Even though no instruments with large footprint and precise alignment are required 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. The comparison between CCD camera-based sensing scheme and detector-based sensing scheme can be found in Table 2.1.

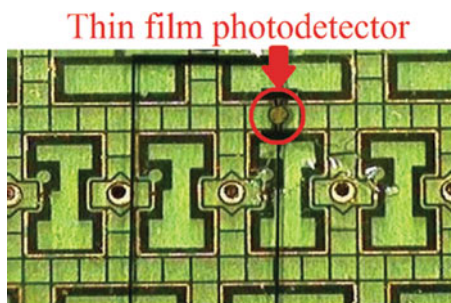


Fig. 2.4 Coplanar digital microfluidic chip with integrated thin film photodetector [15]

Table 2.1 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 of the faulty operation be triggered at the end
Application for photosensitive samples/reagents	Cannot be used	Can be used
Cost	The price of the CCD camera is around \$3,000	The cost of fluorescent labels is around \$30/ μ mol and the Laser is around \$200

2.2.2 “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 [11] 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 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. The unused 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.

Based on the error-recovery mechanism proposed in [11], the strategy for reliability-driven error-recovery will be introduced in Sect. 2.3, and the algorithm for dynamic re-synthesis of error-recovery will be described in Sect. 2.4.

2.2.3 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 desktop computer, thus the control software can interpret the feedback signals. The second interface converts the output data generated by the

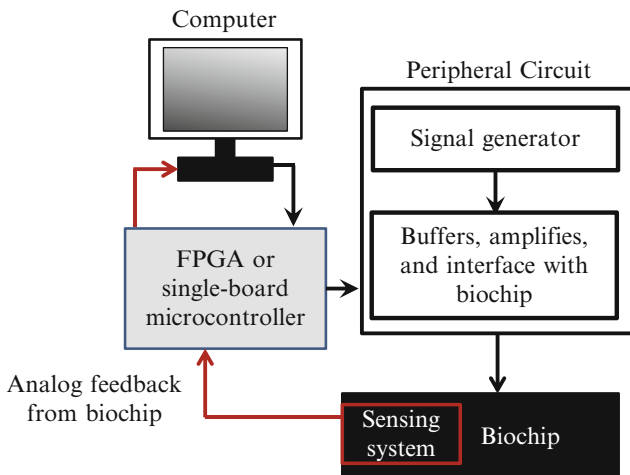


Fig. 2.5 The schematic of the cyberphysical digital microfluidic system. Software running on the computer and the biochip are coupled by a peripheral circuit and a field-programmable gate array (or a single-board microcontroller)

control software to voltage signals that can be directly applied to the electrodes of the biochip.

Figure 2.5 shows a schematic representation of the cyberphysical system. The system consists of a computer, a peripheral circuit, a single-board microcontroller, and the biochip. During the execution of a bioassay, the software running on the computer sends control signals to the biochip. The software at the same time receives feedback from the sensing system on the biochip through the single-board microcontroller. The control software recomputes the schedule of fluidic operations, module placement, and droplet pathways depending on sensor feedback. This process is referred to as reconfiguration and the key idea is to use an on-chip sensing system for concurrent monitoring of experiments. When an error is detected by on-chip sensors, error propagation is prevented by the immediate discarding of the abnormal droplet. The biochip implements error-recovery operations in subsequent operations.

The closed-loop integration in cyberphysical microfluidics can also be used to control the completion time of bioassays. For example, when measuring the glucose in blood, serum samples need to be well-mixed with an enzymatic reagent [19]. During the mixing procedure, the status of the droplet is monitored by an image sensor. The extent to which mixing has been completed can be quantified by analyzing and comparing images of the droplet. Therefore, the control software will force the biochip to continue mixing until the feedback information shows that the droplets are sufficiently well-mixed. Hence without knowing the precise mixer execution time in advance, the mixing operation can be precisely controlled, and the on-chip measurement results for glucose concentration can be as precise

as the results derived by a traditional bench-top analyzer used by a laboratory technician [19].

2.3 Reliability-Driven Error-Recovery

2.3.1 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. It is important to note that, the time cost for adding detection operations is negligible, as the response time of on-chip sensors are in the scale of picoseconds or nanoseconds [13].

On a microfluidic biochip, there are two categories of fluid-handling operations: 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 low and no additional droplets need to be consumed.

The error-recovery process for nonreversible operations is more complicated. In order to re-execute 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 Fig. 2.6a, 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. The following strategies are taken in our approach to reduce the incidence of the worst case:

- For a splitting operation, if only one of its output droplets is used as the input for the immediate successors, the other (redundant) droplet will be stored as a backup for possible error-recovery at a subsequent stage. For example, operation 7 in Fig. 2.6a 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.

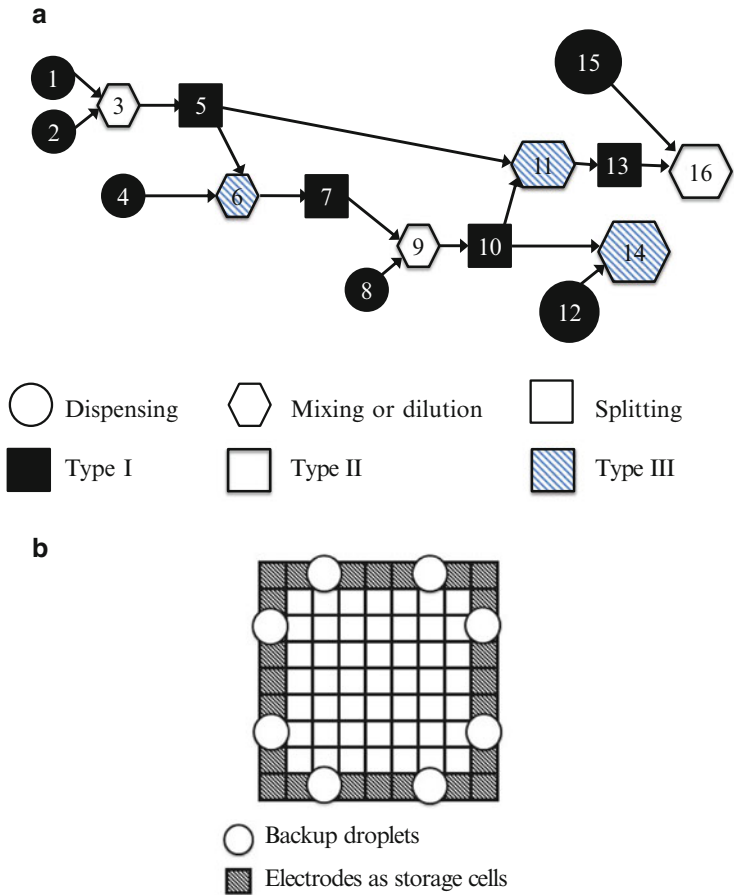


Fig. 2.6 (a) An example of a sequencing graph corresponding to a bioassay protocol; (b) the layout of a biochip with reserved area for error-recovery

- 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. After the bioassay is completed, those unused backup droplets will be sent back to their 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 the biochip. If the answer is yes, then the time cost for this operation can be shortened. Otherwise, more operations will be executed during error-recovery. Based on the above discussion, the operations in the bioassay can be divided into three categories according to the number of operations and droplet consumptions in their error-recovery processes, as shown in Fig. 2.6a.

The fluidic-handling 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 the values of its in-number and out-number:

- If in-number of opt_k is equal to zero, then opt_k is a dispensing operation. Thus we have: $opt_k \in \text{Category I}$.
- 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: $opt_k \in \text{Category I}$.
- 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 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: $opt_k \in \text{Category II}$; otherwise, $opt_k \in \text{Category III}$.

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 . For operations in Category I and II, they can be simply re-executed when an error occurs, as 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. Assume that 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 to enlarge \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 $\mathcal{O} = \{opt_{o_1}, opt_{o_2}, \dots, opt_{o_k}\}$, we define the operator \mathcal{P}_r as:

```

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. 2.7 Pseudocode for determining the recovery operation for opt_i

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

For any operation opt_i , its set of error-recovery operations \mathcal{R}_i can be derived by the procedure presented in Fig. 2.7. According to the above discussion, for any operation opt_i we can derive the set of recovery operations \mathcal{R}_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 Fig. 2.6b; all electrodes on the boundary of the chip are allocated and reserved as storage cells. Thus backup droplets can be easily transported on the biochip.

2.3.2 Reliability Consideration in Error-Recovery

When an error is detected during the execution of a bioassay, it is inefficient to ensure reliable operations by simply re-executing the operation for which an error occurred. 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 discussed below to illustrate the errors caused by the charge-trapping phenomenon and DNA fouling.

When the electrodes of a digital microfluidic biochip are actuated excessively, physically-trapped charge and residual charge may lead to reliability problems [7,8]. Charge trapping is a phenomenon in which charge is trapped and concentrated in the dielectric insulator of the biochip. The trapped charge can lead to a reduction in the electrowetting force and malfunctions in the execution of the bioassay. An example is shown in Fig. 2.8a.

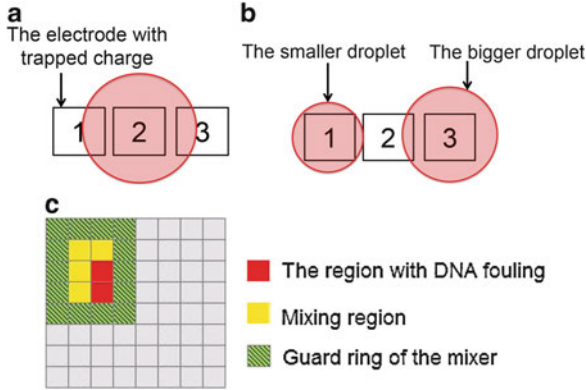


Fig. 2.8 (a) An error caused by the phenomenon of charge trapping; (b) splitting operation with droplets with unbalanced volumes; (c) an error caused by DNA fouling on the surface of a biochip

Suppose Electrode 1 has trapped charge in its dielectric insulator layer, while Electrode 2 and 3 do not suffer from trapped charge. In order to implement a splitting operation, actuation voltages are applied on Electrode 1 and Electrode 3. However, the charge trapped on Electrode 1 will weaken the electrowetting force. The droplet will be split by unequal forces, and the two resulting droplets may have unequal volumes; see Fig. 2.8b. If we simply re-execute the splitting operation and continue to use Electrode 1, additional errors may result. Even worse, the charge-trapping phenomenon may eventually cause permanent dielectric degradation of the electrode [7, 8]. Thus, in order to ensure the reliability of the biochip, the electrode at which the error occurred must no longer be used to implement fluid-handling operations once an error is detected.

The droplets containing macromolecules (such as DNA) may foul the surface of the electrodes [20]. As a result, droplet concentration can change in undesirable ways. If we continue to use these contaminated electrodes, other droplets may also be contaminated. An example of this is shown in Fig. 2.8c. The region where DNA fouling occurred is used as part of a mixer, and the output droplets of the mixing operation may have abnormal concentrations.

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 the on-chip resources occupied by each operation are recorded by the control software. Thus depending on the error droplet, it is feasible to backtrack to the region where error occurs. We consider all electrodes in this region as the suspicious locations for defects. These electrodes will therefore be bypassed.

Table 2.2 Synthesis results for the bioassay shown in Fig. 2.1a

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)

Table 2.3 Synthesis results for the bioassay shown in Fig. 2.1a

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)

2.3.3 Comparison Between Two Sensing Schemes

The two sensing schemes introduced in Sect. 2.2.1 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 sensing systems. Suppose a splitting operation with unbalanced droplets occurs, as shown in Fig. 2.8. 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 Fig. 2.8b 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.

Next, we use the bioassay shown in Fig. 2.1 to further illustrate the differences of these two sensing schemes.

Suppose the droplets for dispensing operations 1 to 6 in Fig. 2.1a are generated from different dispensing ports. For all the mixing operations of the bioassay shown in Fig. 2.1a, their synthesis results are shown in Table 2.3. The module placement result corresponding to synthesis result in Table 2.3 can be found in Fig. 2.9.

It is important to note that, in Table 2.3, “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 2 rows and 3 columns. Thus the mixer is described as a 3×2 mixer at the location (2, 6).

Suppose that in the operation Mix 3 shown in Table 2.3, the DNA-fouling phenomenon occurs after the operation has been underway for 3 s. For the optical

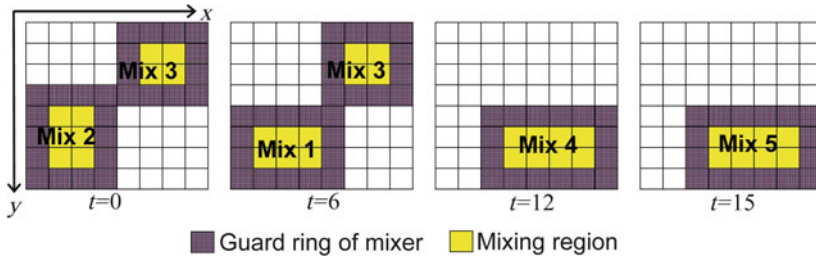


Fig. 2.9 Module placement for the bioassay shown in Fig. 2.1a

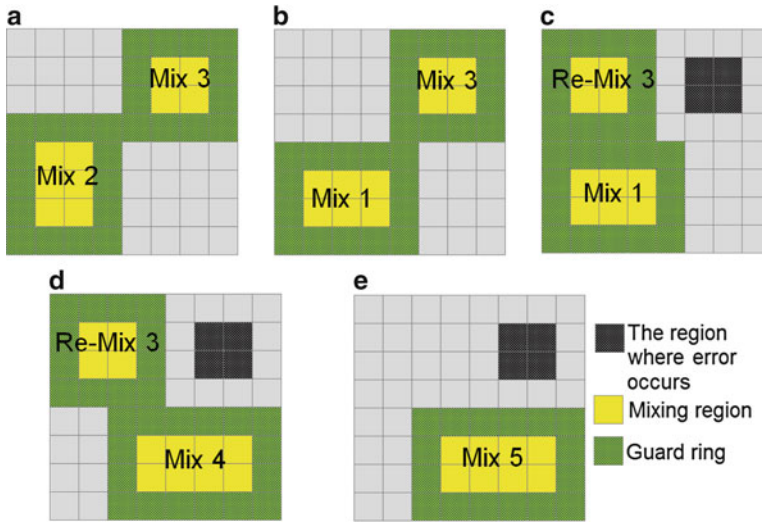


Fig. 2.10 Synthesis results for the bioassay when we use the optical detector-based sensing system. (a) $t = 0$: Mix 2 and Mix 3 begin; $t = 2$: DNA fouling occurs at Mix 3 while it will continue to be executed; $t = 6$: Mix 2 is completed; (b) $t = 6$: Mix 1 begins. Mix 3 is still being executed (even 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

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 Figs. 2.10 and 2.11, respectively. We find that in a detector-based sensing system, recovery can only be triggered at the end of

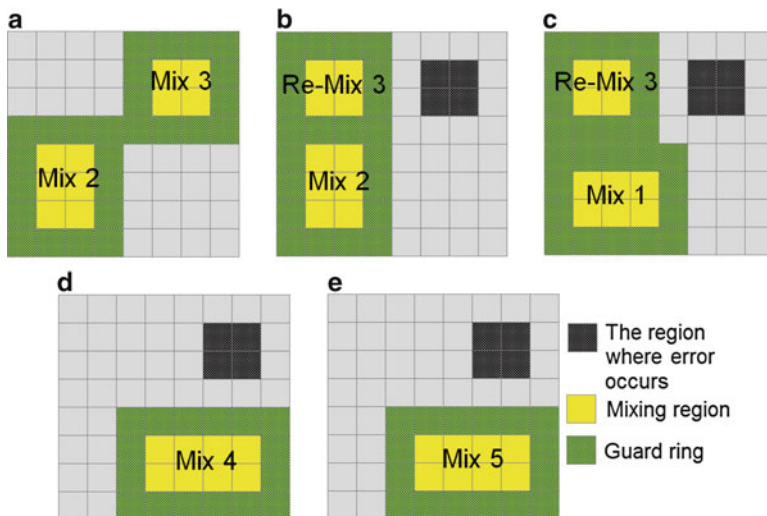


Fig. 2.11 Synthesis results for the bioassay when we use the CCD camera-based sensing system. (a) $t = 0$: Mix 2 and Mix 3 begin; $t = 2$: DNA fouling occurs, Mix 3 stops and will be re-executed; corresponding electrodes are discarded; (b) $t = 2$: recovery operation Re-Mix 3 begins; $t = 6$: Mix 2 is 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

the erroneous operation. While in the CCD camera-based sensing system, recovery can be triggered immediately after an error occurs. On the other hand, in the CCD camera-based sensing system, recovery can only be triggered immediately after an error occurs.

It is important to note that, light from the camera may influence some biochemical substances, e.g., fluorescent markers in the droplet [21]. Thus in order to monitor experiments that include photosensitive samples/reagents, we need to choose the detector-based sensing scheme.

2.4 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 a physical-aware software that can analyze sensor data and dynamically adjust the synthesis result. Adaptations include updates for the schedule of fluid-handling operations, resource binding, module placement, and droplet routing pathways.

The re-synthesis procedure 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.

2.4.1 Off-Line Data Preparation Before Bioassay Execution

The first step in data preparation is converting the sequencing graph of the bioassay to a directed acyclic graph (DAG) and storing the DAG 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. By performing depth-first search on the graph, the predecessors and successors of any operation can be determined [22].

The second step in data preparation is assigning 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 a 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 deriving the initial synthesis result 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:

$$\mathcal{S} = \{M_{opt_1}^*, M_{opt_2}^*, \dots, M_{opt_n}^*\}$$

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

$$M_{opt_i}^* = \langle ts(opt_i), te(opt_i), x(opt_i), y(opt_i), col(opt_i), row(opt_i) \rangle$$

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 (row) is the quantity of columns (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 $M_{opt_i}^*(\tilde{j})$ to represent the \tilde{j} th element in the tuple $M_{opt_i}^*$. For example, the start time of i th operation is written as $M_{opt_i}^*(1)$, the x-coordinate of i th operation is written as $M_{opt_i}^*(3)$, and the number of columns occupied by opt_i is written as $M_{opt_i}^*(5)$.

For simplicity, we use \mathcal{P} to represent the set of all operations in the bioassay; 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 dispensing ports and electrodes) at the same time. Thus opt_w and opt_v must satisfy following constraint:

$$(M_{opt_w}^*(3), M_{opt_w}^*(3) + M_{opt_w}^*(5)) \cap (M_{opt_v}^*(3), M_{opt_v}^*(3) + M_{opt_v}^*(5)) = \emptyset,$$

$$\cup$$

$$(M_{opt_w}^*(4), M_{opt_w}^*(4) + M_{opt_w}^*(6)) \cap (M_{opt_v}^*(4), M_{opt_v}^*(4) + M_{opt_v}^*(6)) = \emptyset,$$

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:

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

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:

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

Previously published computer-aided design methods for digital microfluidic biochips have several proposed algorithms to solve this optimization problem. For example, the PRSA-based synthesis algorithms can be used to quickly derive optimized synthesis results [23].

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.

2.4.2 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.

2.4.2.1 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. The software compares the detection result with a pre-assigned error threshold after each optical detector operation. 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. The software will also bypass all the suspicious electrodes at which the error occurred. For example, assume that 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.

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 comparisons. Thus, error recovery is triggered as soon as an error occurs.

2.4.2.2 Step 2: Update of Sequencing Graph

When an error occurs, the control software determines the required recovery operations. 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. If the error occurs during a reversible operation, the recovery process is simple. If the error occurs during a non-reversible operation, the control software must search the preceding operations until it finds operations that can provide backup droplets to feed the inputs of the recovery subroutine. The pseudocode for the adjustment of the sequencing graph is shown in Fig. 2.12. The definition of error-recovery graph G_{Re_i} for operation opt_i can be found in Sect. 2.3.1. For errors that occur in the operations of Categories I, II, and III, the updated sequencing graphs are shown in Fig. 2.13a–c, respectively. The categorization of operations can be found in Sect. 2.3.1.

-
- 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. 2.12 Pseudocode for adjustment of the sequencing graph

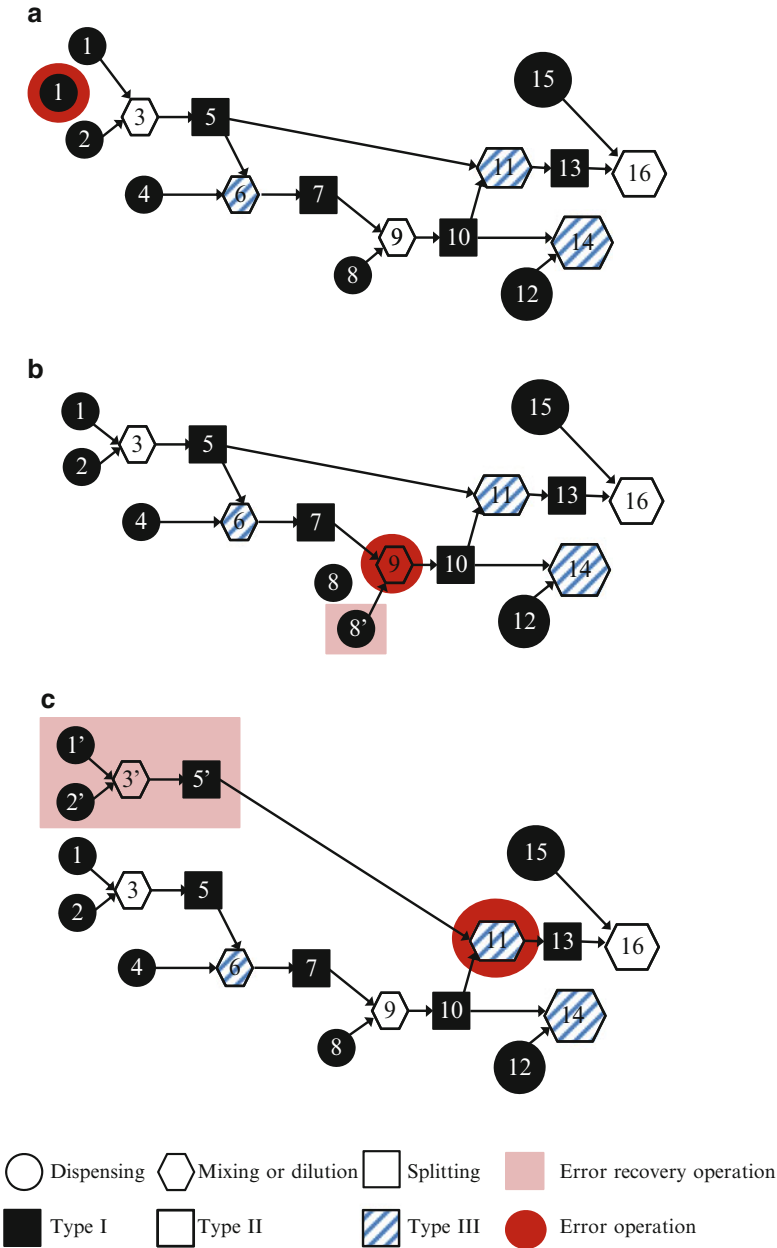


Fig. 2.13 Update of the sequencing graph corresponding to error operations of Categories (a) I; (b) II; and (c) III

It is important to note that, for some operations, the recovery subroutines may change depending on the error. For example, operation 7 in Fig. 2.6a generates two droplets; one of them is used in the subsequent operations 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 ~ 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 additional droplets available at the output of operation 7. Thus the recovery subroutine of operation 14 has to be expanded and it will now include operations 1 ~ 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 performed.

2.4.2.3 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 process of re-synthesis has the following requirements:

- The interruption of other operations should be avoided. Consider the following example. For the bioassay with synthesis results shown in Table 2.3, 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 to avoid the interruption of Mix 1, in the re-synthesis results, the schedule and resource assignment results for Mix 1 should be the same as in the initial synthesis results.
- The electrodes at which an error has been deemed to have occurred should be bypassed in the re-synthesis results.
- The completion time of the bioassay should be minimized.

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 strategy is a PRSA-based global optimization algorithm [1].

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.

Dynamic re-synthesis on the microfluidic array can be modeled as the *module placement with obstacles* problem since the synthesis results for part of the

operations are fixed. Here the operations that are implemented based on the initial synthesis result are fixed a priori as the “obstacles”. The other operations that are necessary for recovery are derived through re-synthesis and they are placed in the remaining available biochip area in a greedy fashion. The detailed steps are described below.

First, based on the topological sort result for operations, the control software places all operations that need to be re-scheduled in a priority queue. 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 allocates on-chip resources to these operations. The 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 $m \times n$ electrode sub-array that is not occupied from current time t to $t + \Delta t$. Here Δt is the time needed for the operation in the $m \times n$ electrode sub-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 binding the resource and determining the start/stop time, the operation will be removed from the priority queue.

Note that when multiple errors are detected at the same time, the above steps can also be used to generate re-synthesis results. 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 are 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 sub-array and P dispensing ports, the computational complexity of looking for available resources (i.e. “the maximum empty rectangle”) in this re-synthesis algorithm is $O(MN + P)$. This is because the software will exhaustively search each electrode/dispensing port in the array and check whether it is available. As the number of dispensing ports can be viewed as constant and we are interested in algorithm scalability for large arrays, the worst-case complexity is $O(MN)$. 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 can be found in Fig. 2.14.

-
- 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: Search available resource for operation q_0 which has the highest priority in Q ;
 - 6: Remove q_0 from Q ;
 - 7: **end while**
-

Fig. 2.14 Pseudocode for dynamic re-synthesis of the bioassay

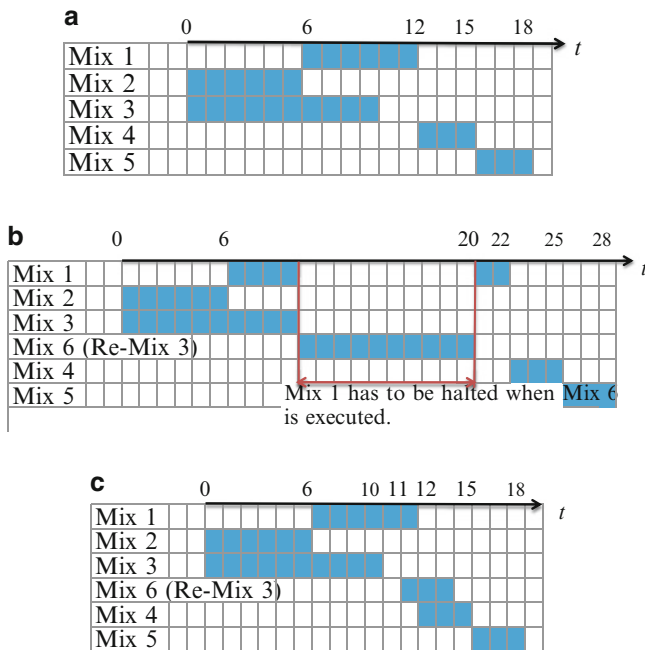


Fig. 2.15 (a) Scheduling result when no error occurs; (b) scheduling when an error occurs in Mix 3. Mix 1 is halted when error operations are executed; (c) scheduling when an error occurs in Mix 3. Here dynamic synthesis strategy is applied at time 10 and error-recovery operations begin at time 11

An example of re-synthesis is shown in Fig. 2.15. Figure 2.15a shows the schedule corresponding to the sequencing graph in Fig. 2.1a. Figure 2.15b, c both show the schedules corresponding to the sequencing graph in Fig. 2.1b. For the sake of clarity, we only show the schedule for mixing operations. Here Fig. 2.15b is the schedule obtained using the error-recovery algorithm of [11].

From Fig. 2.15b, 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. 2.15a increases from 18 time slots to 28 time slots, which can be unacceptable for many applications. The dynamic scheduling result

corresponding to Fig. 2.1b is shown in Fig. 2.15c. When the error is detected in the output of Mix 3 at time 10, the ongoing operation Mix 1 is executed based on the initial synthesis result. We assume that the computing time of generating the new synthesis result is 1 time slot. In practice, the computation time is at least an order of magnitude less than the operation time of fluid-handing operations. Then at time 11, the control software will generate re-synthesis result based on the updated Fig. 2.1b. As shown in Fig. 2.15c, Mix 1 is completed at time 12 without being interrupted. 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 [1]. The inputs and constraints of the re-synthesis problem are different from the initial synthesis problem introduced in Sect. 2.3.1. Suppose the set of operations 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 Sect. 2.3.

We first define an 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 $\mathcal{P}' = \mathcal{P} \cup \mathcal{R}_i \cup \tilde{\mathcal{O}} - \mathcal{P}(t)$. Here \mathcal{R}_i is the set of recovery operations corresponding to erroneous operation opt_i , and $\tilde{\mathcal{O}}$ 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 Sect. 2.4.1. Then based on the module placement information included in initial synthesis result, and locations of electrodes with defects, operations in $\tilde{\mathcal{O}}$ 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 re-synthesis result 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:

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

The above optimization problem can be solved by using the PRSA-based synthesis procedure introduced in [1]. Using this method, we can derive globally-optimized synthesis results with short assay completion time, while the CPU time is in the order of 20 min for a typical bioassay [11]. Thus, this method is not suitable for on-line computation of re-synthesis results.

2.5 Simulation Results

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

2.5.1 Preparation of Plasmid DNA

First, we simulate the bioassay that is called “sample preparation of plasmid DNA by alkalinelysis with SDS” [17, 24]. During sample preparation, a mixture of three reagents is required. The three reagents are:

- R_1 : Alkaline lysis Solution I [50 mM Glucose, 25 mM Tris–HCl (pH 8.0), 10 mM EDTA (pH 8.0)].
- R_2 : Alkaline lysis Solution II [0.2 N NaOH, 1 % SDS (w/v)].
- R_3 : Alkaline lysis Solution III (5 M sodium acetate, glacial acetic acid).

The required concentration of the mixture is 0.22 % of R_1 , 0.44 % of R_2 , and 0.34 % of R_3 , which can be approximated as $\frac{28}{128}$ of R_1 , $\frac{56}{128}$ of R_2 , and $\frac{44}{128}$ of R_3 . Figure 2.16 shows the sequencing graph to obtain the required concentration by mixing R_1 , R_2 , and R_3 . This bioassay is mapped to a 10×10 electrode array and all the electrodes at the boundary of the array are used as storage cells.

When errors are detected, the error-recovery capability of the cyberphysical microfluidic system can be evaluated on the basis of the bioassay completion time. The errors are randomly injected into the chip during the execution of the bioassay and compare the completion time of the two sensing schemes. The results are shown in Fig. 2.17. Here the completion time is derived from the greedy algorithm introduced in Sect. 2.4. The results are the average of the values derived from repeating the experiments ten times. For this case (no error-recovery), the final

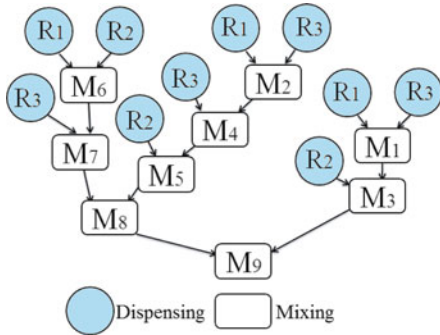


Fig. 2.16 Sequencing graph for sample preparation of plasmid DNA

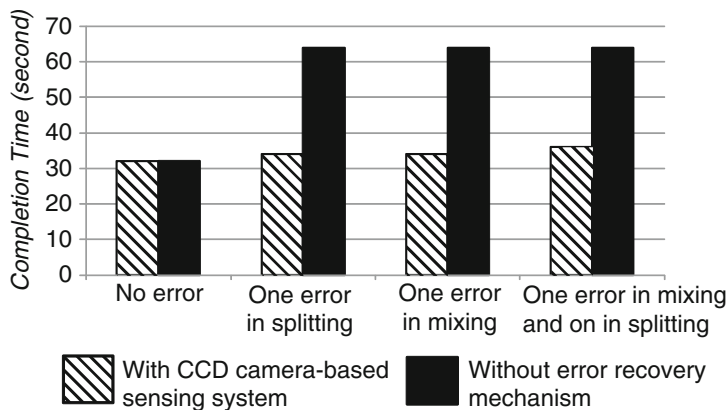


Fig. 2.17 Completion time for biochip with CCD camera-based sensing system, and the biochip without error-recovery mechanism when errors are injected in the sample preparation of plasmid DNA

outcome of the entire experiment will be incorrect if an error occurs during the bioassay. 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.

In reliability-driven error-recovery, the electrodes where an error is deemed to occur, 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 Sect. 2.3.2, these electrodes with defects may further lead to more errors.

To compare the completion times derived from reliability-oblivious and reliability-driven error-recovery procedures, the following simulation is 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 perform 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 2.18 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 4×1 electrode array. As expected, Fig. 2.18 shows that the reliability-driven error-recovery leads to shorter assay completion time in the presence of defects.

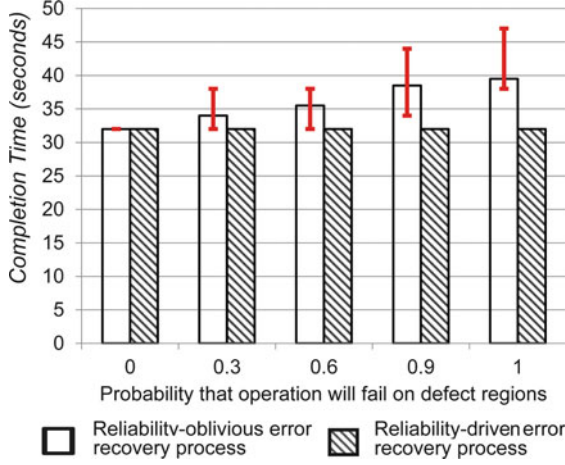


Fig. 2.18 Comparison for the completion time between reliability-driven and reliability-oblivious error-recovery [17] when a 1×4 sub-array is defective in the sample preparation of plasmid DNA. The *error bars* show the maximum and minimum completion time for reliability-oblivious error-recovery in simulation

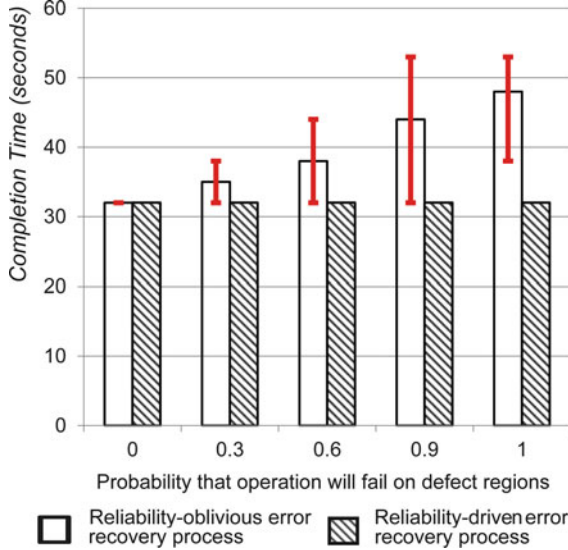


Fig. 2.19 Comparison between the completion time of reliability-driven and reliability-oblivious error-recovery when a 2×4 sub-array is defective in the sample preparation of plasmid DNA. The *error bars* show the maximum and minimum completion time for reliability-oblivious error-recovery in simulation

Next we randomly select another operation as opt_{fe} and run the simulation again. The electrodes that implement opt_{fe} now constitute of a 4×2 electrode array. The simulation results are shown in Fig. 2.19. We find that as expected,

the average completion time for reliability-oblivious error-recovery is higher when more electrodes are defective. The completion time of the reliability-driven error-recovery does not depend on the type of defect on the chip and keeps the minimum completion time.

2.5.2 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. Figure 2.20 shows the sequencing graphs of these two protocols [11]. The protocols for these two bioassays are described in [11].

The completion time of biochips with CCD camera-based sensing systems and without error-recovery mechanism are shown in Fig. 2.21 when errors are injected in the sample preparation of interpolating mixing. The bioassay is mapped to a 10×10 electrode array.

Figure 2.22 reports the completion time when multiple errors are inserted into the interpolating mixing bioassay. Note that the completion time defined here only includes the time spent on fluid-handling operations, and excludes the CPU time consumed on resynthesis. From Fig. 2.22, we see that the completion time achieved by the PRSA-based algorithm and the greedy algorithm are almost the same, but the CPU times for these two algorithms are different. The simulation is performed on a 2.6-GHz, Intel i5 processor with 6 GB of memory. Both re-synthesis algorithms are implemented on the basis of the same initial synthesis result. The CPU time needed is around 33 min for computing the re-synthesis results using PRSA, which was ten times higher than the bioassay completion time; while the CPU time is less than 5 s for the greedy algorithm, which is only 2.5 % of the bioassay completion time. The bioassay completion time derived by the greedy algorithm is only slightly higher for the PRSA. Nevertheless, the greedy algorithm is more suitable for on-line re-synthesis due to the 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 effectiveness for timely bioassay completion. Moreover, the PRSA-based method can be served as the basis for future error-recovery methods based on pre-computation and pre-loading of recovery schedules.

For the exponential dilution protocol introduced in [11], we compare the completion time for the reliability-driven and reliability-oblivious error-recovery methods in Fig. 2.23. 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 performed on a 1×4 electrode sub-array. Then for subsequent operations that are performed on this electrode array with defects, we set P_{fail} as the probability that the operation will fail

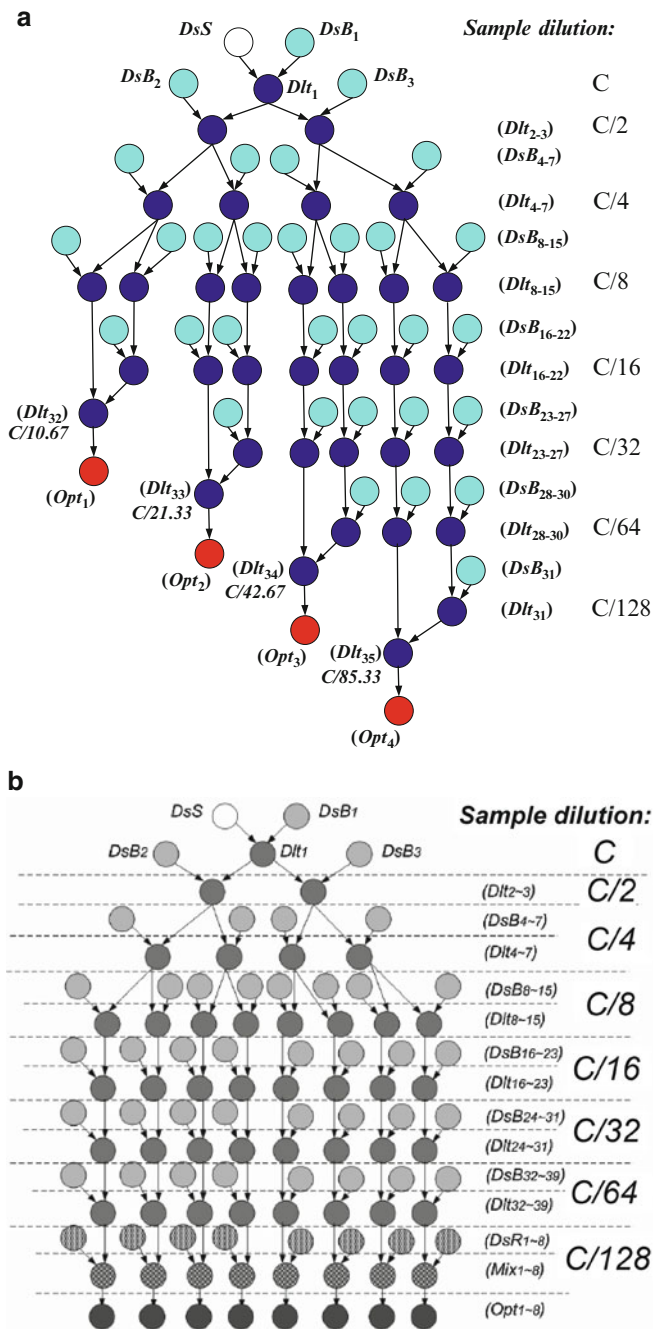


Fig. 2.20 Sequencing graphs for (a) interpolating mixing assay; (b) exponential dilution of a protein sample [11]

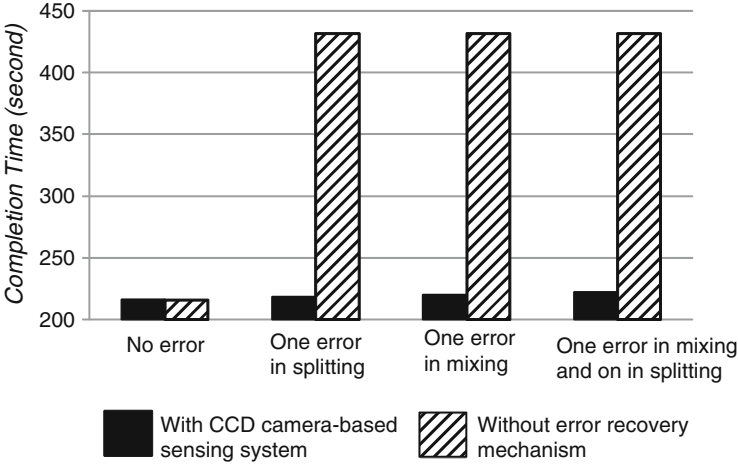


Fig. 2.21 Completion time for biochips with CCD camera-based sensing systems and without error-recovery mechanism when errors are injected in the sample preparation of interpolating mixing

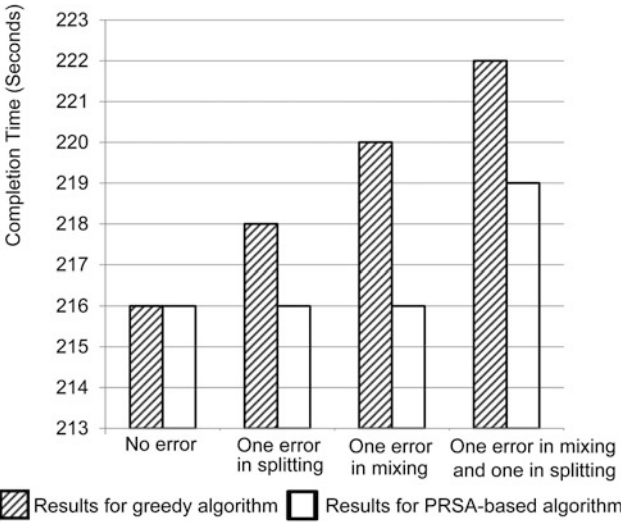


Fig. 2.22 Completion time for the bioassay of interpolating mixing (derived from two re-synthesis algorithms when multiple errors are injected)

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 Fig. 2.23, we find that reliability-driven error-recovery

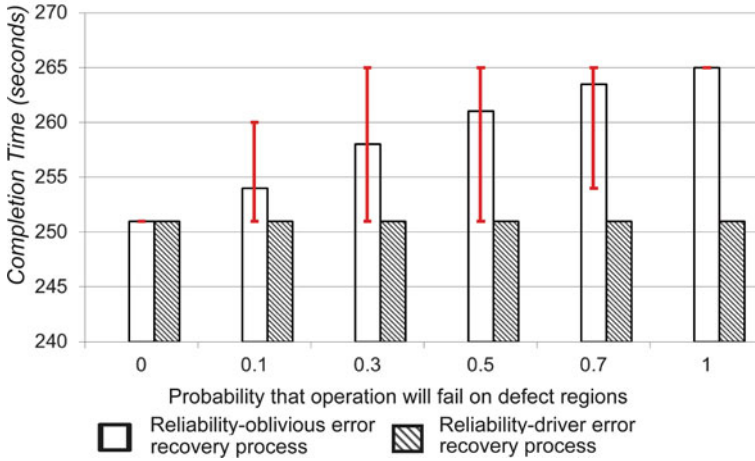


Fig. 2.23 Comparison between the completion time of reliability-driven and reliability-oblivious error-recovery [17] when a 1×4 defect array is injected in exponential dilution. The *error bars* show the maximum and minimum completion time for reliability-oblivious error-recovery in the simulation

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 and the reliability for the experiment is improved. As less reagents/samples are consumed, the cost of the experiment is reduced.

2.6 Chapter Summary and Conclusions

In this chapter, we have shown how recent advances in the integration of a sensing system in a digital microfluidics biochip can be implemented 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 dynamically generate new schedules, module placements, and droplet routing pathways for the bioassay, with minimum impact on the time-to-response. These two methods have been evaluated and 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 to be used as feedback to make decisions about completed operations, to optimize electrode actuation sequences for subsequent operations, and to dynamically reconfigure the biochip. The proposed approach has been evaluated through simulation and its effectiveness demonstrated for three representative protein bioassays.

References

1. K. Chakrabarty and F. Su, *Digital Microfluidic Biochips: Synthesis, Testing, and Reconfiguration Techniques*, Boca Raton, FL: CRC Press, 2006.
2. 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, 2010.
3. E. Maftai, P. Pop, and J. Madsen, "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.
4. 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, 2011.
5. M. Iyengar and M. McGuire, "Imprecise and qualitative probability in systems biology", *International Conference on Systems Biology*, 2007.
6. O. Levenspiel, *Chemical Reaction Engineering*, New York: Wiley, 1999.
7. J. Verheijen and M. Prins, "Reversible electrowetting and trapping of charge: model and experiments", *ACS J. Langmuir*, No. 15, pp. 6616–620, 1999.
8. J. Park, S. Lee, and L. Kanga, "Fast and reliable droplet transport on single-plate electrowetting on dielectrics using nonfloating switching method", *Biomicrofluidics*, vol. 4, Issue. 2, pp. 1–8, 2010.
9. E. Welch, Y.-Y. Lin, A. Madison, and R. Fair, "Picoliter DNA sequencing chemistry on an electrowetting-based digital microfluidic platform", *Biotech. J.*, vol. 6, pp. 165–176, 2011.
10. S. Kotchoni, E. Gachomo, E. Betiku, and O. Shonukan, "A home made kit for plasmid DNA mini-preparation", *African J. Biotech.*, vol. 2, pp. 88–90, 2003.
11. Y. Zhao, T. Xu, and K. Chakrabarty, "Integrated control-path design and error recovery in digital microfluidic lab-on-chip", *ACM JETC*, vol. 3, no. 11, 2010.
12. C. Mein, B. Barratt, M. Dunn, T. Siegmund, A. Smith, L. Esposito, S. Nutland, H. Stevens, A. Wilson, M. Phillips, N. Jarvis, S. Law, M. Arruda, and J. Todd, "Evaluation of single nucleotide polymorphism typing with invader on PCR amplicons and its automation", *Genome Res.*, vol. 10, pp. 330–343, 2000.
13. R. Fair, "Digital microfluidics: Is a true lab-on-a-chip possible?", *Microfluidics and Nanofluidics*, vol. 3, pp. 245–281, 2007.
14. W. Bialek and J. Onuchic, "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.
15. N. Jokerst, L. Luan, S. Palit, M. Royal, S. Dhar, M. Brooke, and T. Tyler II, "Progress in chip-scale photonic sensing", *IEEE Trans. Biomedical Circuits and Sys.*, vol. 3, pp. 202–211, 2009.
16. 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.
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. 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, 2010.
19. B. Hadwen, G. Broder, D. Morganti, A. Jacobs, C. Brown, J. Hector, Y. Kubota, and H. Morgan, "Programmable large area digital microfluidic array with integrated droplet sensing for bioassays", *Lab on a Chip*, pp. 3305–3313, 2012.
20. 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.
21. U. Resch-Genger et. al., "Quantum dots versus organic dyes as fluorescent labels", *Nature Methods*, pp. 763–775, 2008.

22. R. Sedgewick, *Algorithms in C: Graph Algorithms*, Boston, MA: Addison-Wesley, Chapter 23, 2001.
23. S. Kirkpatrick, C. Gelatt and M. Vecchi, "Optimization by simulated annealing", *Science*, vol. 220 (4598), pp. 671–680, May 1983.
24. Y.-L. Hsieh, T.-Y. Ho and K. Chakrabarty, "A reagent-saving mixing algorithm for preparing multiple-target biochemical samples using digital microfluidics", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, pp. 1656–1669, 2012.

Hardware/Software Co-Design and Optimization for
Cyberphysical Integration in Digital Microfluidic Biochips

Luo, Y.; Chakrabarty, K.; Ho, T.-Y.

2015, XII, 197 p. 98 illus., 60 illus. in color., Hardcover

ISBN: 978-3-319-09005-4