Improved Differential Evolution Based on Stochastic Ranking for Robust Layout Synthesis of MEMS Components

Fan, Zhun; Liu, Jinchao; Sørensen, Torben; Wang, P.

Published in:
IEEE Transactions on Industrial Electronics

Link to article, DOI:
10.1109/TIE.2008.2006935

Publication date:
2009

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Improved Differential Evolution Based on Stochastic Ranking for Robust Layout Synthesis of MEMS Components

Zhun Fan, Member, IEEE, Jinchao Liu, Torben Sørensen, and Pan Wang, Member, IEEE

Abstract—This paper introduces an improved differential evolution (DE) algorithm for robust layout synthesis of microelectromechanical system components subject to inherent geometric uncertainties. A case study of the layout synthesis of a comb-driven microresonator shows that the approach proposed in this paper can lead to design results that meet the target performance and are less sensitive to geometric uncertainties than the typical designs. It is also demonstrated that the algorithm proposed in this paper cannot only obtain better results than the standard DE algorithm but also outperform some other state-of-the-art algorithms in constrained optimization.

Index Terms—Differential evolution (DE), microelectromechanical systems (MEMS), robust design, stochastic ranking (SR).

I. INTRODUCTION

MICROELECTROMECHANICAL systems (MEMS) are tiny mechanical devices that are built upon semiconductor chips and are measured in micrometers. They usually integrate a number of functions, including fluidics, optics, mechanics, and electronics, across different physical domains and are used to make numerous devices such as pressure sensors, gyroscopes, engines, accelerometers, etc. Many designs of MEMS are made through engineering experience and back-of-the-envelope calculations and are highly dependent on the knowledge and experience of the designers.

One reason for this is the complexity involved in the modeling, design, and fabrication of MEMS—there are many constraints in designing and fabricating MEMS devices due to the limitations of current fabrication techniques. However, as process technologies become more stable, research emphasis can be shifted from developing specific process technologies toward the design of systems with a large number of reusable components, such as resonators, accelerometers, gyroscopes, and micromirrors. It greatly benefits the MEMS designers if the routine design of frequently used components can be optimized automatically by computer programs, while the designers can take more time in contemplating the more creative conceptual designs.

It has been shown that the performance of individual components influences the quality of the whole system. For example, the frequency stability of a MEMS resonator can directly affect the quality of the MEMS RF system in which it serves as a component of a filter or an oscillator [1]. Because microresonators are basically 2.5 dimensional devices, the design automation of microresonators boils down to a layout synthesis problem, which has been carried out by many researchers. Some notable research uses deterministic numeric approaches [2]–[4] and metaheuristic approaches such as evolutionary computation [5], [6] and simulated annealing [7], [8].

It is notable that, with current micromachining techniques, the fabrication process variation in MEMS is inevitable when devices are miniaturized to the point of process limitations [1]. For example, it is reported in [9] that the width of a typical suspension beam has a fabrication tolerance of about 10%. How to design MEMS that is most insensitive to fabrication process variation is therefore an important issue in MEMS design. This paper also addresses this important issue, i.e., incorporating uncertainty in MEMS design through robust optimization.

Many approaches exist in the literature to incorporate uncertainty in a design formulation. Taguchi [10] introduced the concept of robust design to improve the quality of products with significant variations in their manufacturing process by reducing the sensitivity of the design performance to possible sources of variations without an attempt to eliminate the sources. Robust design has been developed and applied in many areas. Some examples include robust optical coating design [11], robust design of a vibratory microgyroscope [13], an active micromixer [14], and brushless dc motors [15].

In this paper, we present a robust optimization approach for designing MEMS subject to process-induced geometrical uncertainties. In this approach, we first formulate the robust design problem as a multiobjective constrained optimization problem [16], [17] and then solve it using an improved differential evolution (DE). DE is a strong and efficient optimization algorithm capable of handling nonlinear, nondifferentiable, and multimodal objective functions [18]. A case study based on the layout synthesis of comb-driven microresonator shows that the robust designs nominally meet the target performance and are less sensitive to geometric uncertainties. It is also demonstrated...
that the algorithm proposed in this paper cannot only obtain better results than the standard DE algorithm but also outperform some other state-of-the-art algorithms in constrained optimization.

The remainder of this paper is organized as follows. Section II introduces the formulation of the general robust optimization problem that is used to formulate the MEMS layout synthesis problem in this paper. Section III describes the method of modeling uncertainty in MEMS fabrication process. Section IV explains in details the improved DE algorithm (IDE) used to do the robust layout synthesis. Section V gives the description of the case study of comb-driven microresonator design, and Section VI presents experimental results. Section VII concludes this paper with a brief summary.

II. FORMULATION OF THE ROBUST OPTIMIZATION PROBLEM

This paper considers the application of a general robust optimization problem that can be formulated as the following [17]. Let $\bar{x} = \{x_1, x_2, \ldots, x_n\}$ be an array of design variables of a given design problem. We assume that the uncertainty $\delta = \{\delta_1, \delta_2, \ldots, \delta_n\}$ can be characterized as a random vector with the following statistics:

\[
E(\delta) = 0_{n \times 1}
\]

\[
E(\delta \delta^T) = \Omega \in \mathbb{R}^{n \times n} \quad (2)
\]

where $\Omega$ is the covariance matrix and is positive semidefinite. If the uncertainties are uncorrelated, then $\Omega$ is diagonal; otherwise, the off-diagonal entries are nonzero when correlation exists.

Given a function $f(\bar{x}, \delta)$ describing the performance of a design merit, the robust design problem that we aim to solve is to minimize the expected value of the squared error between the actual and target performances. We can write this as

\[
\min \bar{x} \quad E \left( f(\bar{x}, \delta) - \bar{f} \right)^2 \quad \text{subject to} \quad g_i(\bar{x}) \leq 0 \quad (3)
\]

where $\bar{f}$ is the target performance, and the expectation is taken over the random vector $\delta$. In addition, $g_i(\bar{x}) \leq 0$ represents a list of constraints to be satisfied.

The problem posed in (3) is a difficult robust optimization problem to solve in general. To simplify the problem, we choose to approximate $f(\bar{x}, \delta)$ with a first order Taylor series expansion in $\delta$ as

\[
f(\bar{x}, \delta) \approx f(\bar{x}, 0) + \nabla_{\delta} f(\bar{x}, 0) \delta\quad (4)
\]

where $\nabla_{\delta} f(\bar{x}, 0)$ is the gradient of $f(\bar{x}, \delta)$ with respect to $\delta$. Using this approximation, we can expand the expression of $(f(\bar{x}, \delta) - \bar{f})^2$ into

\[
(f(\bar{x}, \delta) - \bar{f})^2 \approx \left( f(\bar{x}, 0) - \bar{f} \right)^2 + 2 (f(\bar{x}, 0) - \bar{f}) \times \nabla_{\delta} f(\bar{x}, 0) \delta + \nabla_{\delta} f(\bar{x}, 0) \delta \delta^T \nabla_{\delta}^T f(\bar{x}, 0). \quad (5)
\]

Taking the expectation of the aforementioned equation, we can get

\[
E \left( \left( f(\bar{x}, \delta) - \bar{f} \right)^2 \right) \approx \left( f(\bar{x}, 0) - \bar{f} \right)^2 + 2 (f(\bar{x}, 0) - \bar{f}) \times \nabla_{\delta} f(\bar{x}, 0) E(\delta) + \nabla_{\delta} f(\bar{x}, 0) E(\delta \delta^T) \nabla_{\delta}^T f(\bar{x}, 0). \quad (6)
\]

By reducing (6), based on our assumptions about the mean and covariance of $\delta$ according to (1) and (2), we obtain

\[
E \left( f(\bar{x}, \delta) - \bar{f} \right)^2 \approx \left( f(\bar{x}, 0) - \bar{f} \right)^2 + \nabla_{\delta} f(\bar{x}, 0) \Omega \nabla_{\delta}^T f(\bar{x}, 0). \quad (7)
\]

Substituting the approximation in (7) back into the original design problem posed in (3) yields

\[
\min \bar{x} \quad \left\{ \left( f(\bar{x}, 0) - \bar{f} \right)^2 + \nabla_{\delta} f(\bar{x}, 0) \Omega \nabla_{\delta}^T f(\bar{x}, 0) \right\}
\]

\[
\text{subject to } g_i(\bar{x}) \leq 0. \quad (8)
\]

To normalize the cost function, we decide to divide through by $\bar{f}^2$. We then refer to the following expression as our robust design problem:

\[
\min \bar{x} \quad \left( \left( \frac{f(\bar{x}, 0) - \bar{f}}{\bar{f}} \right)^2 + \frac{1}{\bar{f}^2} \left( \nabla_{\delta} f(\bar{x}, 0) \Omega \nabla_{\delta}^T f(\bar{x}, 0) \right) \right)
\]

\[
\text{subject to } g_i(\bar{x}) \leq 0. \quad (9)
\]

It is now easy to see that the expression we want to minimize has two distinct terms. For notational convenience, we will label the two terms as

\[
N(\bar{x}) \equiv \left( \frac{f(\bar{x}, 0) - \bar{f}}{\bar{f}} \right)^2 \quad (10)
\]

\[
D(\bar{x}, \Omega) \equiv \frac{1}{\bar{f}^2} \nabla_{\delta} f(\bar{x}, 0) \Omega \nabla_{\delta}^T f(\bar{x}, 0). \quad (11)
\]

With the aforementioned definitions, the robust design problem posed in (9) becomes

\[
\min \bar{x} \quad \left\{ N(\bar{x}), D(\bar{x}, \Omega) \right\} \quad \text{subject to } g_i(\bar{x}) \leq 0. \quad (12)
\]

The first term $N(\bar{x})$ penalizes the deviation of the nominal solution $f(\bar{x}, 0)$ from the target $\bar{f}$, while the second term $D(\bar{x}, \Omega)$ penalizes the sensitivity of the design with respect to $\delta$. The first term is a performance index, while the second term is a robustness index. Since there are two objectives in the formation of the cost function to be minimized, a tradeoff is usually needed to be made by the designer. In practice, if we want to obtain robust designs, a simple way is to sum the two objectives into a single one and take it as the objective to be minimized. One disadvantage of applying this method in our work is that it cannot guarantee that $D(\bar{x}, \Omega)$ is reduced after optimization. Theoretically, it is possible that $D(\bar{x}, \Omega)$ increases if the decrease of $N(\bar{x})$ is more than the increase of $D(\bar{x}, \Omega)$ within the reduced sum. In our approach, we set a small threshold constraint $N(\bar{x}) \leq 1.0 \times 10^{-6}$ and then focus on minimizing the value of $D(\bar{x}, \Omega)$. 

Authorized licensed use limited to: Danmarks Tekniske Informationscenter. Downloaded on October 28, 2009 at 05:10 from IEEE Xplore. Restrictions apply.
Fig. 1. Under- and overetch of a MEMS structure.

III. MODELING UNCERTAINTY IN MEMS FABRICATION PROCESS

We assume that the uncertainty in the fabrication process is introduced by etch- or lithograph-induced variations in linewidth, and the structure is etched uniformly.

Fig. 1 shows the two uniform etch scenarios on a structure—overetch and underetch. Take the underetch situation, for example, after process variation is introduced, some design variables may increase (such as L1 and L2) and other design variables may decrease, while some others may stay unchanged (such as L4).

We can model the geometric process variations using a simple additive uncertain model

\[ \bar{x} = \bar{x} + \delta \]  

where \( \bar{x} \) is the uncertain (actual) design vector, and for the aforementioned simple example, \( \delta = \{ \delta_{L1}, \delta_{L2}, \delta_{L3}, \delta_{L4} \} \).

Since the structure is etched uniformly, if we define \( \rho \) to be a normal random variable with zero mean and standard deviation of \( \sigma \), then we can write

\[ \delta = \rho \zeta \]  

where \( \zeta = [1, 1, -1, 0]^T \) and is called a variation vector. Note that, in the condition of underetch, L1 and L2 increase, L3 decreases, and L4 is not changed. Also note that, in this case, \( \rho \) is positive; the aforementioned facts can easily be verified by (14).

Because \( \rho \) is a normal random variable, it can also be used to model the overetch situation, in which \( \rho \) will take a negative value.

According to (2), we can obtain

\[ \Omega = E(\delta^T \delta) = \sigma^2 \zeta \zeta^T. \]

IV. ROBUST OPTIMIZATION USING IDE

Many types of evolutionary optimization approaches have been developed and implemented as design optimization tools [19], [20]. Genetic algorithms with a robust solution searching scheme were first presented by Tsutsui and Ghosh [21] and later discussed by Deb and Gupta [22], [23]. Jin and Branke [24] made a thorough survey of applying evolutionary computation in uncertain environments. Most recently, Lee et al. [25] use robust evolutionary algorithms for unmanned (combat) aerial vehicle aerodynamics and radio cross section design optimization. One advantage of using genetic algorithms is its convenience to solve the optimization problem with both discrete and continuous design variables. While it is very difficult for many numerical optimization approaches (for example, gradient-based approaches) to include considerations of feature size constraints in MEMS design [4], it is quite convenient for genetic algorithms to do so. We need to modify the objective function only slightly, mapping real values of design variables to integer multiples of the feature size before using them in the formulations of constraints and objectives. No modifications to the genetic algorithm are needed. In this paper, we always set the feature size as 0.09 \( \mu \)m. It is also very convenient for evolutionary computation to integrate integer design variables such as the number of comb fingers used in a microresonator.

An IDE based on Stochastic Ranking (IDE-SR) was developed and used to solve the robust layout synthesis problem in this paper. The succeeding sections will first introduce the standard DE algorithm and then explain the novel mechanisms developed in IDE-SR in details.

A. Standard DE

DE is one of the most recent evolutionary algorithms for solving real-parameter optimization problems. In each iteration, DE creates one new offspring individual by combining one parental individual and the differences of several other individuals in the same population. The generated offspring individual replaces the parental individual only if it is better. In general, DE has three parameters that can impact its performance significantly: scaling factor \( F \), crossover control parameter \( p_{CR} \), and population size \( N_p \).

The population of DE contains \( N_p \) \( n \)-dimensional individuals

\[ \bar{x}_{i,G} = \{ x_{i,1,G}, x_{i,2,G}, \ldots, x_{i,n,G} \}, \quad i = 1, 2, \ldots, N_p \]  

where \( G \) denotes the generation number. Because it is considered beneficial to the search process if the initial population can be statistically evenly distributed over the entire search space, each variable of all individuals in the initial population is randomly decided by a uniform distribution between lower and upper bounds predefined for each variable.

At each generation, a target vector \( \bar{x}_{i,G} \) is first selected randomly, and then, a mutant vector \( \bar{v}_{i,G} \) is created by disturbing the target vector using a mutation operation; after that, a trial vector \( \bar{u}_{i,G} \) is formed by applying crossover operation between the target and mutant vectors. Finally, a selection operation is executed between the trial and target vectors to decide which vector goes to the next generation. The procedure is repeated \( N_p \) times to create all individuals for an offspring generation. The main procedure for DE is shown in Fig. 2 and explained in detail as follows.

1) Mutation Operation: For each target vector \( \bar{x}_{i,G} \) at generation \( G \), an associated mutant vector \( \bar{v}_{i,G} = \{ v_{i,1,G}, v_{i,2,G}, \ldots, v_{i,n,G} \} \) can be created by using one of the mutation strategies. The most commonly used strategies are as follows.
1. Initialize
   Parameters: $N_p, P_{CR}, F, t, m$, where $N_p$ denotes the size of population; $P_{CR}$ denotes the probability of crossover, $F$ is a scaling factor, $t$ denotes the generation, and $m$ represents the number of individuals participating in mutation operation (e.g., 3 or 5).
   Generate the initial generation $P_0, t = 1$.
2. While termination criteria not satisfied do:
   3. Evaluate population $P_t = \{x_1^t, x_2^t, ..., x_{N_p}^t\}$.
   4. For $k = 1$ to $N_p$ do
      5. Select $x_{r_1}^t, x_{r_2}^t, x_{r_3}^t \in P_t$ at random.
      6. $u_k^t \leftarrow \text{mutate}(x_{r_1}^t, x_{r_2}^t, x_{r_3}^t, F)$;
      7. $v_k^t \leftarrow \text{crossover}(u_k^t, x_{r_4}^t, P_{CR})$.
      8. If $v_k^t$ is better than $x_k^t$ then
         9. $x_k^t \leftarrow v_k^t$;
      10. End if
   11. End for
   12. $t \leftarrow t + 1$;
   13. End while

Fig. 2. Pseudocode of the iterative search procedure of DE.

where the indexes $r_1, r_2, r_3, r_4,$ and $r_5$ represent the random and mutually different integers generated between 1 and $N_p$, and $\bar{x}_{best,G}$ is the best individual at generation $G$: 
- “rand/1” $\bar{v}_{i,G} = \bar{x}_{r_1,G} + F(\bar{x}_{r_2,G} - \bar{x}_{r_3,G})$;
- “best/1” $\bar{v}_{i,G} = \bar{x}_{best,G} + F(\bar{x}_{r_1,G} - \bar{x}_{r_2,G})$;
- “current to best/1” $\bar{v}_{i,G} = \bar{x}_{i,G} + F(\bar{x}_{best,G} - \bar{x}_{i,G}) + F(\bar{x}_{r_1,G} - \bar{x}_{r_2,G})$;
- “best/2” $\bar{v}_{i,G} = \bar{x}_{best,G} + F(\bar{x}_{r_1,G} - \bar{x}_{r_2,G}) + F(\bar{x}_{r_3,G} - \bar{x}_{r_4,G})$;
- “rand/2” $\bar{v}_{i,G} = \bar{x}_{r_1,G} + F(\bar{x}_{r_2,G} - \bar{x}_{r_3,G}) + F(\bar{x}_{r_4,G} - \bar{x}_{r_5,G})$.

Different strategies have different features for different applications. It is also possible to use a combination of two or more strategies to better cope with certain application.

2) Crossover Operation: After mutation, a “binary” crossover operation is applied to form the final trial vector $\bar{u}_{i,G}$, according to its corresponding target vector $\bar{x}_{i,G}$ and mutant vector $\bar{v}_{i,G}$:

$$\bar{u}_{i,j,G} = \begin{cases} \bar{v}_{i,j,G}, & \text{if } \text{rand} \leq P_{CR} \text{ or } j = j_{\text{rand}} \\ \bar{x}_{i,j,G}, & \text{otherwise} \end{cases}$$

(17)

where $i = 1, 2, ..., N_p$; $j = 1, 2, ..., n$; and index $j_{\text{rand}}$ is a randomly chosen integer within the range $[1, n]$. By making use of $j_{\text{rand}}$, it can be guaranteed that the trial vector $\bar{u}_{i,G}$ will differ from its target vector $\bar{x}_{i,G}$ by at least one parameter.

3) Selection Operation: After evaluating the target vector $\bar{x}_{i,G}$ and the corresponding trial vector $\bar{u}_{i,G}$, a “knockout” competition is played between them, and the vector with smaller objective function value is selected and added to the next population

$$\bar{x}_{i,G+1} = \begin{cases} \bar{u}_{i,G}, & \text{if } f(\bar{u}_{i,G}) \leq f(\bar{x}_{i,G}) \\ \bar{x}_{i,G}, & \text{otherwise} \end{cases}$$

(18)

Because each individual has both the values of objective function and constraint violation for comparison, it is important to use some rules for the purpose of comparison. According to our empirical experience, different rules of handling constraints used can actually lead to very different results in constrained optimization algorithms.

B. Different Rules of Handling Constraints

Very few constraint handling techniques have been reported in DE. Two very important and similar techniques are

Fig. 3. Pseudocode of SR [29].

Fig. 4. Illustration of the modified mutation strategy in IDE-SR. Note that the population ranked by SR is divided into upper part Q1 and lower part Q2. The difference of one randomly selected individual $r_2$ from Q1 and one randomly selected individual $r_3$ from Q2 forms a differential vector pointing toward $r_2$.
Fig. 5. Pseudocode of the iterative search procedure of the IDE based on SR: IDE-SR.

1. **Initialize:**
   Parameters: $N_p, p_f, \mu, \sigma, p_{CR}, \gamma$; where $N_p$ denotes the size of population; $p_f$ is a parameter used in stochastic ranking, $\mu, \sigma$ denote the mean and variance of normal distribution, respectively; $p_{CR}$ denotes the probability of crossover, $\gamma$ represents the number of individuals in the upper part of the population $Q_1$.

2. Generate the initial generation $P_0$.
3. **While** termination criteria not satisfied do:
4. Evaluate population $P_t$: $(f, \phi) = \text{eval}(P_t)$; where $f, \phi$ denote objective and violation of constraints, respectively.
5. Rank population using stochastic ranking: $I = \text{stochastic_rank}(f, \phi, p_f)$;
6. Divide population into two sets:
   - $Q_1^t = \{x_{1(1)}^t, x_{1(2)}^t, \ldots, x_{1(\gamma)}^t\}$;
   - $Q_2^t = \{x_{1(\gamma+1)}^t, x_{1(\gamma+2)}^t, \ldots, x_{1(N_p)}^t\}$;
7. For $k = 1$ to $N_p$ do:
8. Select $x_{1}^t, x_{2}^t \in Q_1^t$ at random;
9. Select $x_{1}^t \in Q_2^t$ at random;
10. $u_{k}^t \leftarrow x_{1}^t + N(\mu, \sigma) \times (x_{2}^t - x_{1}^t)$;
11. $v_{k}^t \leftarrow \text{crossover}(u_{k}^t, x_{1}^t, p_{CR})$;
12. If $v_{k}^t$ is better than $x_{k}^t$ then $x_{k}^t \leftarrow v_{k}^t$;
13. End if
14. End for
15. $t \leftarrow t + 1$;
16. End while

IDE-SR: An Improved DE Based on SR

The "rand/1" mutation strategy used in the standard DE provides no information of direction toward the global optimum. If the information of direction can be obtained and utilized in the search process, the performance of the algorithm has a potential to be improved. To avoid the search to be stuck in local minimum, however, the direction information should not

It is worthwhile to point out that both approaches bear some resemblance with an approach proposed by Deb [28] previously, even though Deb’s approach is not based in DE. The key difference also lies in the comparison for the case of two infeasible individuals: Lampinen’s method makes the comparison in the Pareto sense, Deb’s method sums all the constraint violations and compares a single value, and Becerra and Coello’s method makes normalization for the constraint violations before summing them together.

Like the selection of mutation strategies, the selection of proper constraint handling techniques is highly dependent on applications. In this paper, Becerra and Coello’s approach was selected because it outperformed the others.

C. IDE-SR: An Improved DE Based on SR

The "rand/1" mutation strategy used in the standard DE provides no information of direction toward the global optimum. If the information of direction can be obtained and utilized in the search process, the performance of the algorithm has a potential to be improved. To avoid the search to be stuck in local minimum, however, the direction information should not

proposed by Lampinen [26] and Becerra and Coello [27]. Both techniques use three rules for the replacement during the selection procedure, and the first two are the same. They are as follows.

1) A feasible individual is always better than an infeasible individual.
2) If both individuals are feasible, the one with better value of the objective function is selected for the next generation.

The third rule, regarding the situation when both individuals are infeasible, is different. In Lampinen’s approach, the comparison is made in the Pareto sense in the constraint violation space. It can be expressed as follows.

1) If both individuals are infeasible, the parent is replaced if the new individual has lower or equal violation for all the constraints.

In Becerra and Coello’s approach, a sum of normalized constraint violations is used for comparison and can be written as follows.

1) If both individuals are infeasible, the individual with less level of constraint violations is better. The level of constraint violation is measured with the normalized constraints with the expression of $\text{viol}(x_j) = \sum_{c=1}^{\text{const}}(g_c(x)/g_{\text{max},c})$, where $g_c(x)$ denotes the violated constraints of the problem and $g_{\text{max},c}$ is the largest violation of the constraint $g_c(x)$ found so far.
be local but global. To define a “global direction” information for
the whole population is not an easy task, particularly when
each individual has actually two features to compare with the
others in a constraint optimization problem—one feature is the
objective value, and the other is the level of constraint violation.
How to optimally balance them in the comparison procedure
presents a challenge.

SR [29] provides a convenient and powerful mechanism to
balance the dominance in ranking the whole population with
both objective value and constraint violation as comparison
criteria. The pseudocode of SR is provided in Fig. 3.

The IDE-SR is designed with a focus on a modified mutation
strategy, which can be described in more details as the follow-

For the generation of trial vectors, the whole population
is first made to undergo an SR procedure. Then, the ranked
population is divided into two parts—upper and lower parts.
The upper part comprises of the “better” individuals who have
been ranked high after the SR procedure. For each individual
trial vector, the upper part contributes two “good” randomly
selected individuals, and the lower part contributes one ran-
domly selected individual that is “less good.” The three indi-
viduals then make a mutation operation according to “rand/1”
strategy, with the difference vector obtained through extracting
one “good” individual with the “less-good” individual. It is
notable that, in this way, the difference vector will always be
directed toward the upper part of the population, thus leading
the population to search upwards (Fig. 4). This procedure is
repeated until the whole population of trial vectors is obtained.
The rest of the algorithm is almost the same as the standard
DE, with the exception that the scaling factor $F$ can become
a random variable as a variation of the algorithm. The overall
procedure of the IDE-SR algorithm can be illustrated using the
pseudocode listed in Fig. 5.

V. CASE STUDY

A case study in the area of MEMS design was carried
out to verify the effectiveness of the aforementioned robust
optimization method using the IDE-SR. The design problem is
a comb-drive microresonator with 15 mixed-type design vari-
able and 24 design constraints, both linear and nonlinear. The
following section gives a more detailed description of the case
study.

A. Comb-Driven Microresonator Design

The comb-driven microresonator problem was originally
taken from [4]. The goal of the design is to robustly match
the resonant frequency to the predefined target frequency in
the presence of geometric process variations. The comb-drive
microresonator is fabricated in a polysilicon surface microstruc-
tural process. The layout of the comb-driven microresonator is
shown in Fig. 6(a) and can be specified by 15 design variables,
as shown in Fig. 6(b) [4]. The vector of design variables can
then be defined as follows:

$$\vec{x} = [L_b, w_b, L_t, w_t, L_{sy}, w_{sy}, L_{cy}, w_{cy}, L_c, w_c, L_{sa}, w_{sa}, x_o, V, N_c]$$

where

- $L_b$ and $w_b$: the length and width of flexure beam, respectively;
- $L_t$ and $w_t$: the length and width of truss beam, respectively;
- $L_{sy}$ and $w_{sy}$: the length and width of shuttle yoke, respectively;
- $L_{cy}$ and $w_{cy}$: the length and width of comb yoke, respectively;
- $L_c$ and $w_c$: the length and width of comb fingers, respectively;
- $w_{sa}$: the width of shuttle axle;
- $g$: the gap between comb fingers;
- $x_o$: the comb finger overlap;
- $V$: the voltage amplitude;
- $N_c$: the number of rotor comb fingers.

It is noted that the first 13 design variables have units of
micrometers. They are discrete variables because they can only
be integer multiples of the feature size, which is set to be
0.09 µm in this paper. The fourteenth design variable has units
of volts and is a continuous variable. The fifteenth variable has no unit and is an integer variable.

The constraints for the design variables are also listed as follows: $2 \leq L_s \leq 400$, $2 \leq w_b \leq 20$, $2 \leq L_t \leq 400$, $2 \leq w_c \leq 20$, $2 \leq L_{sy} \leq 400$, $10 \leq w_{sy} \leq 400$, $10 \leq w_{sa} \leq 400$, $10 \leq w_{cy} \leq 400$, $2 \leq L_c \leq 700$, $8 \leq L_e \leq 400$, $2 \leq w_e \leq 20$, $2 \leq L_{ea} \leq 400$, $4 \leq x_b \leq 400$, $0 \leq V \leq 50$, and $3 \leq N_c \leq 50$.

In addition, we assume that $w_c = g = d$ in our design, for the special case of equal comb finger width, gap, and spacing above the substrate. Some design variables are predefined: They are $w_{ba} = 11$, $w_{ca} = 14$, $\gamma = 4$, and $t = 2$, in which $w_{ba}$ is the width of beam anchors, $w_{ca}$ is the width of stator comb, and $t$ is the thickness of the microresonator.

There are a number of design constraints to be considered for the comb-driven microresonator cell component, including both geometric and functional constraints. In this paper, without loss of generality, we consider the following 24 constraints:

$$g_1(x) : -(L_{cy} + 2g + 2w_c) \leq 0$$
$$g_2(x) : L_{cy} + 2g + 2w_c - 700 \leq 0$$
$$g_3(x) : -(L_{sy} + 2L_b + 2w_s) \leq 0$$
$$g_4(x) : L_{sy} + 2L_b + 2w_s - 700 \leq 0$$
$$g_5(x) : -(3L_t + w_{sy} + 4L_e - 2x_b + 2w_{cy} + 2w_{ca}) \leq 0$$
$$g_6(x) : 3L_t + w_{sy} + 4L_e - 2x_b + 2w_{cy} + 2w_{ca} - 700 \leq 0$$
$$g_7(x) : L_c - (x_0 + x_{disp}) - 200 \leq 0$$
$$g_8(x) : 4 - L_c + (x_0 + x_{disp}) \leq 0$$
$$g_9(x) : -(2N_c + 1)w_c + 2N_c g - L_{cy} \leq 0$$
$$g_{10}(x) : (2N_c + 1)w_c + 2N_c g - L_{cy} - 700 \leq 0$$
$$g_{11}(x) : -(x_0 - x_{disp} - 4) \leq 0$$
$$g_{12}(x) : x_0 - x_{disp} - 200 \leq 0$$
$$g_{13}(x) : 4 - (L_t - x_{disp}) - (W_{sy} + W_b)/2 \leq 0$$
$$g_{14}(x) : L_t - x_{disp} - (W_{sy} + W_b)/2 - 200 \leq 0$$
$$g_{15}(x) : 2 - (L_{sy} - 2W_{ba} - W_{sa})/2 \leq 0$$
$$g_{16}(x) : (L_{sy} - 2W_{ba} - W_{sa})/2 - 200 \leq 0$$
$$g_{17}(x) : 2 - x_{disp} \leq 0$$
$$g_{18}(x) : x_{disp} - 100 \leq 0$$
$$g_{19}(x) : 5 - Q \leq 0$$
$$g_{20}(x) : Q - 1e5 \leq 0$$
$$g_{21}(x) : -x_{disp}/L_b \leq 0$$
$$g_{22}(x) : x_{disp}/L_b - 0.1 \leq 0$$
$$g_{23}(x) : -K_{cy}/K_y \leq 0$$
$$g_{24}(x) : K_{cy}/K_y - 1/3 \leq 0$$.

Among them, the first sixteen are linear constraints, and the last eight are nonlinear constraints

$$x_{disp} = QF_{e,x}/K_x$$  \hspace{1cm} (20)
TABLE II
RESULTS OF NONROBUST LAYOUT SYNTHESIS OF COMB-DRIVEN MICRORESONATOR

<table>
<thead>
<tr>
<th>RUN NO.</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_n (\mu m)$</td>
<td>91.71</td>
<td>155.09</td>
<td>164.34</td>
<td>126.63</td>
<td>123.39</td>
<td>170.73</td>
<td>168.48</td>
<td>129.78</td>
<td>104.76</td>
<td>163.35</td>
</tr>
<tr>
<td>$w_n (\mu m)$</td>
<td>4.05</td>
<td>5.22</td>
<td>8.46</td>
<td>5.67</td>
<td>4.68</td>
<td>8.82</td>
<td>11.88</td>
<td>5.4</td>
<td>4.59</td>
<td>5.31</td>
</tr>
<tr>
<td>$L_w (\mu m)$</td>
<td>42.84</td>
<td>44.37</td>
<td>29.7</td>
<td>62.37</td>
<td>90.36</td>
<td>47.7</td>
<td>40.32</td>
<td>29.61</td>
<td>59.4</td>
<td>46.89</td>
</tr>
<tr>
<td>$w_w (\mu m)$</td>
<td>4.32</td>
<td>8.91</td>
<td>15.39</td>
<td>14.58</td>
<td>16.38</td>
<td>5.31</td>
<td>15.12</td>
<td>12.33</td>
<td>9</td>
<td>11.7</td>
</tr>
<tr>
<td>$L_m (\mu m)$</td>
<td>253.5</td>
<td>121.5</td>
<td>127.98</td>
<td>146.52</td>
<td>164.97</td>
<td>56.41</td>
<td>199.53</td>
<td>141.21</td>
<td>375.84</td>
<td>77.4</td>
</tr>
<tr>
<td>$w_m (\mu m)$</td>
<td>18.81</td>
<td>17.1</td>
<td>18.63</td>
<td>12.15</td>
<td>16.38</td>
<td>24.48</td>
<td>44.37</td>
<td>19.09</td>
<td>10.62</td>
<td>19.08</td>
</tr>
<tr>
<td>$w_m (\mu m)$</td>
<td>25.83</td>
<td>24.21</td>
<td>38.88</td>
<td>54.99</td>
<td>32.67</td>
<td>20.61</td>
<td>37.62</td>
<td>16.92</td>
<td>19.35</td>
<td>12.87</td>
</tr>
<tr>
<td>$w_m (\mu m)$</td>
<td>62.64</td>
<td>78.21</td>
<td>17.37</td>
<td>96.57</td>
<td>13.95</td>
<td>48.6</td>
<td>21.87</td>
<td>28.98</td>
<td>38.25</td>
<td>56.88</td>
</tr>
<tr>
<td>$L_m (\mu m)$</td>
<td>338.2</td>
<td>231.57</td>
<td>395.1</td>
<td>300.48</td>
<td>377.64</td>
<td>300.03</td>
<td>554.94</td>
<td>319.5</td>
<td>452.16</td>
<td>560.25</td>
</tr>
<tr>
<td>$L_m (\mu m)$</td>
<td>70.02</td>
<td>103.14</td>
<td>76.5</td>
<td>65.07</td>
<td>48.42</td>
<td>43.65</td>
<td>44.19</td>
<td>64.26</td>
<td>105.03</td>
<td>74.07</td>
</tr>
<tr>
<td>$w_m (\mu m)$</td>
<td>5.4</td>
<td>7.2</td>
<td>4.77</td>
<td>4.86</td>
<td>5.67</td>
<td>3.69</td>
<td>2.79</td>
<td>4.23</td>
<td>4.5</td>
<td>8.19</td>
</tr>
<tr>
<td>$L_m (\mu m)$</td>
<td>71.1</td>
<td>103.14</td>
<td>351.9</td>
<td>147.33</td>
<td>16.83</td>
<td>123.57</td>
<td>224.82</td>
<td>135.09</td>
<td>66.42</td>
<td>82.08</td>
</tr>
<tr>
<td>$x_c (\mu m)$</td>
<td>19.26</td>
<td>65.79</td>
<td>28.44</td>
<td>35.01</td>
<td>34.65</td>
<td>33.57</td>
<td>34.2</td>
<td>27.81</td>
<td>79.2</td>
<td>24.03</td>
</tr>
<tr>
<td>$V (volt)$</td>
<td>47.96</td>
<td>46.18</td>
<td>43.25</td>
<td>44.96</td>
<td>42.92</td>
<td>44.56</td>
<td>44.34</td>
<td>47.66</td>
<td>47.48</td>
<td>46.57</td>
</tr>
<tr>
<td>$N_c$</td>
<td>31</td>
<td>22</td>
<td>56</td>
<td>51</td>
<td>56</td>
<td>51</td>
<td>93</td>
<td>43</td>
<td>58</td>
<td>21</td>
</tr>
<tr>
<td>$f_0 - 200kHz$</td>
<td>8.55e-4</td>
<td>5.99e-4</td>
<td>2.49e-4</td>
<td>3.30e-4</td>
<td>3.24e-4</td>
<td>3.85e-4</td>
<td>3.34e-4</td>
<td>1.57e-4</td>
<td>2.20e-4</td>
<td>1.29e-4</td>
</tr>
</tbody>
</table>

In other words, in this particular case study, the definition of $f(\bar{x}, 0)$ in (10) can be expressed as

$$ f(\bar{x}, 0) = \omega_n(\bar{x}, 0) $$

(32)

where $\bar{f}$ can be predefined by users. In this paper, without loss of generality, $\bar{f} = 200$ kHz.

VI. EXPERIMENTS

As shown in (10) and (11), there are two design objectives to minimize in the robust design problem. The first objective relates to the design performance, while the second objective reflects the robustness of the design. To verify that involving the robustness consideration in the optimization process can help to reduce the sensitivity of the resulting designs to the variations of the design variables, we carry out a comparative study. In the first set of runs of IDE-SR, we only consider the first design objective, i.e., the performance objective. In the second set of runs of IDE-SR, we consider both performance and robustness objectives.

To verify that the performance of IDE-SR is competitive, a comparison study was also made among IDE-SR, standard DE, and two other state-of-the-art approaches in constrained evolutionary approaches.

A. Results of Nonrobust Layout Synthesis

In the first set of runs, we only considered the performance objective $f_{obj} = N(\bar{x})$, as described in (10). Ten runs of experiments using IDE-SR algorithm were repeated with $\bar{f} = 200$ kHz. The parameters of the constrained genetic algorithm are listed in Table I.

The experimental data were obtained in Table II. It is noted that ten results represent ten different designs that all satisfy the design constraints and have natural frequencies very closely matching to the target $\bar{f} = 200$ kHz.

B. Results of Robust Layout Synthesis

For robust layout synthesis of the comb-driven microresonator, we need to consider both objectives in (10) and (11). To calculate the robustness index in (11), we need to know the variation vector according to (15). By examining the layout schematic of the comb-driven microresonator, we found that the variation vector can be set as follows:

$$ \xi = [0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ -1 \ 1 \ 0 \ 0] $$

According to (15), to obtain the robustness index in (11), we also need to make an assumption about $\sigma$. In this paper, we assume $\sigma = 0.1 \mu m$.

In the robust layout synthesis, we took the robustness index as the optimization objective $f_{obj} = D(\bar{x}, \Omega)$. In addition, another constraint $N(\bar{x}) \leq 1.0e^{-6}$ is added to the constraint list. Ten runs of experiments using IDE-SR algorithm were repeated, with the same parameters defined in Table I. The experimental data were listed in Table III. It can be seen from Tables II and III that the values of the objective $D(\bar{x}, \Omega)$ are smaller in the case of robust designs than those in the case of nonrobust designs. The next section demonstrates that the reduced objective values of $D(\bar{x}, \Omega)$ lead to more robust designs.
TABLE III

<table>
<thead>
<tr>
<th>RUN NO.</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>L₀ (μm)</td>
<td>305.73</td>
<td>323.37</td>
<td>316.44</td>
<td>329.76</td>
<td>294.66</td>
<td>274.77</td>
<td>320.4</td>
<td>306.63</td>
<td>299.97</td>
<td>290.88</td>
</tr>
<tr>
<td>wₓ (μm)</td>
<td>15.75</td>
<td>20.07</td>
<td>16.56</td>
<td>19.80</td>
<td>19.62</td>
<td>17.10</td>
<td>20.07</td>
<td>17.82</td>
<td>15.84</td>
<td>15.57</td>
</tr>
<tr>
<td>L₁ (μm)</td>
<td>198.36</td>
<td>201.78</td>
<td>134.91</td>
<td>190.17</td>
<td>199.71</td>
<td>181.71</td>
<td>201.33</td>
<td>188.37</td>
<td>133.38</td>
<td>121.59</td>
</tr>
<tr>
<td>wᵧ (μm)</td>
<td>14.49</td>
<td>2.07</td>
<td>15.48</td>
<td>2.07</td>
<td>2.07</td>
<td>9</td>
<td>2.07</td>
<td>2.07</td>
<td>15.21</td>
<td>14.85</td>
</tr>
<tr>
<td>L₂ (μm)</td>
<td>59.49</td>
<td>47.79</td>
<td>36.09</td>
<td>36.09</td>
<td>104.85</td>
<td>119.34</td>
<td>53.64</td>
<td>67.86</td>
<td>69.57</td>
<td>88.29</td>
</tr>
<tr>
<td>wᵧ (μm)</td>
<td>18.27</td>
<td>10.08</td>
<td>202.95</td>
<td>42.30</td>
<td>16.47</td>
<td>70.47</td>
<td>10.53</td>
<td>50.4</td>
<td>60.57</td>
<td>112.32</td>
</tr>
<tr>
<td>wₓ (μm)</td>
<td>33.48</td>
<td>15.93</td>
<td>10.08</td>
<td>10.08</td>
<td>67.32</td>
<td>93.33</td>
<td>27.36</td>
<td>34.47</td>
<td>43.47</td>
<td>31.50</td>
</tr>
<tr>
<td>L₁ (μm)</td>
<td>10.44</td>
<td>10.08</td>
<td>12.51</td>
<td>11.43</td>
<td>10.08</td>
<td>10.08</td>
<td>10.17</td>
<td>10.08</td>
<td>47.16</td>
<td>15.12</td>
</tr>
<tr>
<td>L₂ (μm)</td>
<td>652.68</td>
<td>644.94</td>
<td>657.27</td>
<td>253.35</td>
<td>314.28</td>
<td>617.67</td>
<td>607.41</td>
<td>636.93</td>
<td>649.89</td>
<td>649.44</td>
</tr>
<tr>
<td>xₓ (μm)</td>
<td>12.06</td>
<td>12.06</td>
<td>12.06</td>
<td>12.06</td>
<td>12.06</td>
<td>12.15</td>
<td>12.06</td>
<td>29.34</td>
<td>12.15</td>
<td></td>
</tr>
<tr>
<td>L₁ (μm)</td>
<td>398.07</td>
<td>399.51</td>
<td>382.59</td>
<td>29.07</td>
<td>104.85</td>
<td>20.34</td>
<td>239.94</td>
<td>27.99</td>
<td>244.08</td>
<td>96.03</td>
</tr>
<tr>
<td>xₓ (μm)</td>
<td>6.03</td>
<td>6.03</td>
<td>6.03</td>
<td>6.03</td>
<td>6.03</td>
<td>6.12</td>
<td>6.03</td>
<td>6.03</td>
<td>6.03</td>
<td>6.03</td>
</tr>
<tr>
<td>V (volt)</td>
<td>50.00</td>
<td>50.00</td>
<td>50.00</td>
<td>50.00</td>
<td>50.00</td>
<td>49.99</td>
<td>50.00</td>
<td>49.99</td>
<td>49.99</td>
<td>49.99</td>
</tr>
<tr>
<td>N₁ (10⁶)</td>
<td>33</td>
<td>27</td>
<td>36</td>
<td>22</td>
<td>26</td>
<td>33</td>
<td>27</td>
<td>24</td>
<td>38</td>
<td>37</td>
</tr>
<tr>
<td>D(Ω, Ω)</td>
<td>1.691e-3</td>
<td>9.277e-4</td>
<td>1.786e-3</td>
<td>2.011e-3</td>
<td>2.197e-3</td>
<td>2.201e-3</td>
<td>2.100e-3</td>
<td>9.682e-4</td>
<td>1.935e-3</td>
<td>2.096e-3</td>
</tr>
</tbody>
</table>

**C. Comparison of Robust and Nonrobust Results**

It is noted that, in the robust design process, we minimized the robustness objective. To verify that doing this helps the resulting designs to increase their insensitivity to geometric uncertainties, we designed a comparative study as the following: We put two designs in one group for comparison by selecting one design from the robust design group and the other from the nonrobust design group. We then ran Monte Carlo simulations to model uncertain MEMS fabrication processes. We introduced the same variations to the design variables of both designs to emulate uniform overetch and/or underetch situations. To represent the variations in the process, we generated 10,000 Gaussian random vectors with a standard deviation of 0.1 μm. The natural frequencies of both the robust and nonrobust designs were calculated, and the histograms of them were plotted, as shown in Fig. 7.

According to Fig. 7, we can see that robust design has a much tighter distribution of natural frequencies and therefore is much less sensitive to geometric variations. The tests of other design candidates from both robust and nonrobust design groups revealed similar results. Figs. 8 and 9 drawn with SUGAR [30] show the layout of two exemplar nonrobust and robust designs, respectively.

**D. Comparison of Different Optimization Algorithms**

A comparison study was also made to compare the performance of IDE-SR with the standard DE and two other state-of-the-art evolutionary constraint optimization algorithms, which are improved stochastic ranking based evolutionary strategy (ISRES) [29] and nondominated sorting genetic algorithm.
(NSGA)-II [31]. For each algorithm, 50 independent runs were carried out, with the best, mean, worst, and standard deviation of the obtained results that are all recorded in Table IV for comparison purposes. Bold values in Table IV indicate the best results among different algorithms.

It is clear from Table IV that IDE-SR outperforms DE, ISRES, and NSGA-II in terms of best, mean, and worst results. The most important criterion to be compared is the best result, because usually, we choose the design vector related to the best result to make the design. It is also important to note that DE, NSGA-II, and IDE-SR all performed very stably and could successfully find feasible solutions that satisfy all the constraints every time out of 50 independent runs. Due to their stochastic nature, evolutionary algorithms cannot guarantee convergence every time. However, the aforementioned three algorithms show very good consistency in this particular problem. ISRES, however, failed to do so in five times out of 50 independent runs. It is also notable that, if we use a population size of 100 in ISRES, it could not find a feasible solution. The reported results for ISRES were obtained with a population size of 200, which is double the population size used in other algorithms.

Fig. 10 shows the curves of objective values versus generation number recorded in one exemplar evolutionary process of both algorithms—IDE-SR and standard DE. It can be seen that IDE-SR has a stronger capability to find better objective values.

TABLE IV

<table>
<thead>
<tr>
<th></th>
<th>DE</th>
<th>ISRES</th>
<th>NSGA-II</th>
<th>IDE-SR</th>
</tr>
</thead>
<tbody>
<tr>
<td>best</td>
<td>1.74×10^{-3}</td>
<td>9.68×10^{-4}</td>
<td>1.00×10^{-3}</td>
<td>9.27×10^{-4}</td>
</tr>
<tr>
<td>mean</td>
<td>2.01×10^{-3}</td>
<td>1.69×10^{-3}</td>
<td>1.61×10^{-3}</td>
<td>1.56×10^{-3}</td>
</tr>
<tr>
<td>worst</td>
<td>2.45×10^{-3}</td>
<td>2.24×10^{-3}</td>
<td>2.94×10^{-3}</td>
<td>1.82×10^{-3}</td>
</tr>
<tr>
<td>SR</td>
<td>50(50)</td>
<td>45(50)</td>
<td>50(50)</td>
<td>50(50)</td>
</tr>
<tr>
<td>PS</td>
<td>100</td>
<td>200</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

SR: success rate; PS: population size

VII. CONCLUSION

Layout synthesis is an important stage for a structured design of MEMS [32], [33], after the stage of the system-level design [34]. This paper has developed a novel constrained optimization algorithm IDE-SR, which is an IDE based on SR, and reports a method of robust layout synthesis of MEMS based on it. The method transforms the robust design problem into a multiobjective constrained optimization problem and then solves it by using IDE-SR. Simulation results based on a case study of the layout synthesis of a comb-driven microresonator show that the
design solutions obtained using the method proposed in this paper are much less sensitive to process-induced uncertainties. This paper has also shown that the IDE-SR algorithm cannot only obtain better results than the standard DE algorithm but also outperform some other state-of-the-art evolutionary constrained optimization algorithms.

The next step is to combine the detailed design described in this paper with system-level design so that a hierarchical design procedure can be automated. In addition, another work of testing the performance of IDE-SR on a comprehensive set of benchmark problems [35] is being conducted.

ACKNOWLEDGMENT

The authors would like to thank E. Goodman for his valuable comments and suggestions in revising and finalizing the manuscript. The authors would also like to thank Z. Cai and Y. Wang for their general discussions on the DE algorithm when they visited the Technical University of Denmark as Research Scholars during April–August 2007. The visit was gratefully sponsored by the Otto Mønsted Fund of Denmark.

REFERENCES


Fig. 10. Curves of objective value versus generation of both DE and IDE-SR.
Zhuo Fan (S'01–M'04) received the B.S. and M.S. degrees in control engineering from Huazhong University of Science and Technology, Wuhan, China, in 1995 and 2000, respectively, and the Ph.D. degree in electrical and computer engineering from Michigan State University, East Lansing, in 2004. From 2004 to 2007, he was an Assistant Professor with the Department of Mechanical Engineering, Technical University of Denmark, Lyngby, Denmark. He is currently an Associate Professor with the Department of Management Engineering, Technical University of Denmark. His research interests include MEMS, mechatronics, robotics, design automation and optimization, evolutionary computation, computational intelligence, computer vision, and intelligent transportation and power system.

Jinchao Liu received the B.S. and M.S. degrees in control science and engineering from Wuhan University of Technology, Wuhan, China, in 2004 and 2007, respectively. He is currently working toward the Ph.D. degree in the Department of Management Engineering, Technical University of Denmark, Kongens Lyngby, Denmark. He is also with FORCE Technology, Brøndby, Denmark. His main research interests include computer vision, image processing, welding automation, microelectromechanical systems, robotics control, and evolutionary computation.

Pan Wang (S’01–M’08) received the B.S. degree in industrial automation from Wuhan University of Technology, Wuhan, China, in 1994, and the M.S. and Ph.D. degrees in systems engineering from Huazhong University of Science and Technology, Wuhan, China, in 1998 and 2003, respectively. He is an Associate Professor and the Director of the Laboratory of Control and Decision, School of Automation, Wuhan University of Technology. He has published over 35 journal papers, three monographs, and 25 conference papers. His research interests include intelligent control and optimization, decision analysis, and biomedical intelligent information systems. Dr. Wang has been the recipient of about ten academic or teaching awards in China.

Torben Sørensen received the M.S. and Ph.D. degrees in robotics and control engineering from the Technical University of Denmark (DTU), Kongens Lyngby, Denmark, in 1986 and 1996, respectively. He was an Assistant Professor of control and engineering design with DTU from 1996 to 1998, where he was an Associate Professor in the Department of Mechanical Engineering from 2001 to 2007 and an Associate Professor in the Department of Management Engineering in 2008. He has participated as a Work Group Leader in three international scientific projects and authored or coauthored several scientific papers, as well as teaching materials including lecture notes, etc. His main research interests include robotics control, welding automation, image processing, and mechanical systems. Dr. Sørensen was nominated for the title “Lecturer of the Year at DTU” in 2003. He was a committee member of the Danish Industrial Robot Association (in 2004–2008). He tragically passed away in 2008.