An Auto-Adjusting Hybrid Quantum Genetic Algorithm-Spectre Platform for the Multi-objective Optimization of Analog Circuit Sizing

Thinh Quang Do[a], Hoang Trong Nguyen[b,c], Bradley D. E. McNiven[a], Trang Hoang[b,c,*], Lihong Zhang[a], Octavia A. Dobre[a], Trung Q. Duong[a,d]

[a] *Department of Electrical and Computer Engineering, Faculty of Engineering and Applied Science, Memorial University of Newfoundland, St. John's, A1B 3X9, NL, Canada*

[b] *Department of Electronics, Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam*

[c] *Vietnam National University Ho Chi Minh City, Linh Trung Ward, Thu Duc City, Ho Chi Minh City, Vietnam*

[d] *School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast, United Kingdom*

Abstract

Among the phases constituting analog circuit design, circuit sizing is considered labor-intensive, formidable, and heavily experience-dependent due to its non-linearity. As a result, design automation coupled with effective optimization techniques has arisen as a feasible candidate to address challenges with circuit design and satisfy the increasing need for high-performance circuits. Among evolutionary algorithms, the combination of the genetic algorithm (GA) and quantum computing techniques has yielded the hybrid quantum genetic algorithm (HQGA) which has proven to be an effective optimization method in many fields due to its convergence rate and near-optimal solutions. This paper introduces an upgraded version of HQGA we call the Auto-adjusting Hybrid Quantum Genetic Algorithm (AHQGA) which avoids premature convergence and improves convergence speed through the use of an additional best-fitness-based scheme for rotation angles. In particular, this work proposes the utility of AHQGA for the multi-objective optimization

[*] Corresponding author: Trang Hoang (e-mail: hoangtrang@hcmut.edu.vn).

of analog circuit sizing, with the two-stage Miller-compensated operational amplifier (op-amp) used as a topological case study. Additionally, for an objective evaluation, optimization results by AHQGA are compared with those by HQGA with fixed rotation angles and classical GA.

Keywords: Auto-adjusting hybrid quantum genetic algorithm, classical genetic algorithm, hybrid quantum genetic algorithm, rotation angles, two-stage Miller-compensated operational amplifier

1. Introduction

The design of analog integrated circuits (ICs) mainly consists of three phases: topological-level design, circuit sizing (or parameter-level design) and layout extraction [1, 2]. In the circuit sizing phase, due to the MOSFET's characteristics and multidimensional trade-offs between various analog performance metrics, the relationship between design parameters and circuit variables is highly non-linear [3]. As a result, analog circuit design has been considered a complicated and time-consuming process which depends heavily on designer experience [4]. In order to reduce this reliance as well as fulfill the increasing demand for high-performance circuits, analog design automation has arisen as a promising research area that captures the attention of academia [5]. Regarding implementation, circuit sizing should be categorized as a non-linear constrained and multi-objective optimization [3]. As a result, before solving any optimization problem, determining the method, algorithm as well as interface for the optimization process is considered essential.

Automated sizing methods for analog circuits can be mainly categorized into equation-based and simulation-based methods [2]. In the former case, equation-based methods utilize circuit analysis to derive posynomial or monomial equations for targeted performance metrics. Despite fast execution time and a high chance of global optimum assurance, obtaining such expressions is viewed as laborious and demands high designer effort, especially for complex circuits [6]. Furthermore, closed-form and explicit functions for circuit parameters often require simplifications and approximations, which leads to the sacrifice of MOSFET's higher-order effects and a lack of precision for the model [2]. In contrast, the simulation-based counterpart utilizes real-time simulation data to evaluate the performance of a specific circuit, process objective functions (or fitness functions), and design constraints in the form of black-box functions. This helps the optimization process explore the search space in an oriented manner with a high degree of accuracy [2, 7]. This method is considered both flexible and convenient when compared with equation-based optimization due to its adaptability to any circuit topology and design specifications with a short set-up time [8], and the optimization method of choice used in this work.

With the optimization method chosen, an appropriate optimization algorithm should be then selected. Among numerous existing model-free optimization methods, the genetic algorithm (GA), based on the Darwinian principle of natural selection and concepts of natural genetics, is commonly used as an effective solution for large search spaces [2]. However, the ability of GA to update its population is not oriented to the current best individual, but in a highly random manner with a selection, crossover and mutation step instead. To overcome this issue in our research, the auto-adjusting hybrid quantum genetic algorithm (AHQGA) is introduced as an inter-discipline algorithm inheriting GA's characteristics and quantum computing concepts [9]. Rather than expressing chromosomes in binary bits as GA, AHQGA encodes chromosomes by quantum bits and updates them by quantum gates [10]. Thanks to this encoding scheme, AHQGA can guide the exploration of the whole quantum population in the search space toward the currently best chromosomes. This might result in the large diversity of population, rapid convergence and high global search capability of AHQGA [11].

Moreover, it is worth noticing two key features of our AHQGA that might address the shortcomings of several other works. Firstly, our work utilizes the random coefficient $p(\alpha)$ for quantum measurement. Regarding observation or measurement of quantum bits, [11] and [12] performed it with a constant coefficient that practically deactivates the probabilistic mechanism of quantum measurement, which in turn might degrade the algorithm's performance. Thus, to preserve the randomness of observation, our implementation is similar to that presented in [13, 10] with $p(\alpha)$, which is further elucidated in section 3. Secondly and more importantly, our research utilized an auto-adjusting rotation scheme relative to the difference between the currently considered chromosome of the population and the global best one. This differentiates from the fixed rotation angles when updating quantum chromosomes as in conventional hybrid quantum genetic algorithm (HQGA) of [13, 14]. As a result, premature convergence can be prevented and AHQGA's efficiency can be improved to its greatest extent. In summary, thanks to its noticeable advantages, AHQGA, with two aforementioned solutions to tackle existing research gaps, is nominated as the algorithm for our optimization process, which is virtually one of the leading scientific works on quantum-based optimization for analog circuit sizing.

Last but not least, determining an optimization interface for the connection between the simulation-based method and AHQGA is crucial. In the work presented in [15, 6], HSPICE was used as the circuit simulator for design parameters' value. Nevertheless, an extra step is typically needed to gather necessary data by employing scripting languages [2]. The Spectre simulator meanwhile enables the incorporation of the SKILL programming language's syntax within Ocean-based scripts, which does not require that step [2]. Considering the Spectre simulator's role in the whole optimization system, the adaptability of SKILL programming reflects the authors' preference for Spectre compared to its HSPICE counterpart in terms of manipulating output data.

Being one of the fundamental blocks in analog IC design, two-stage Miller-compensated op-amp is chosen as the typical case study that might demonstrate the groundwork for quantum-inspired circuit sizing optimization in the design automation of analog circuits. With only 100 iterations of AHQGA, the optimized op-amp achieved a fitness value of 0.5654 at the 49^{th} iteration, which is chosen as

$$\text{Fitness} = \frac{\text{UGB} \times C_L}{I_{total}} \times \frac{tan(\text{PM})}{tan(PM_{REF})}, \tag{1}$$

where UGB is the unity gain-bandwidth product, C_L is the load capacitor at the output node, I_{total} is the total current, PM is the phase margin and PM_{REF} is the reference phase margin [16]. The achieved results are promising and comparable in terms of the fitness value and number of iterations to achieve that optimal value. More significantly, considering the same set-up and initial conditions, optimization results by AHQGA have been compared with those by HQGA with fixed rotation angles and GA for an objective assessment.

To sum up, the following are the technical contributions provided by this paper:

1. The proposal of the AHQGA-Spectre model for the efficient optimization of analog circuit sizing. This not only sets the foundation for quantum-inspired optimization of analog circuit sizing but also overcomes the issues of global optimum and convergence speed of traditional algorithms.

2. The utilization of the Spectre simulator as the interface between the AHQGA optimization algorithm and simulation-based optimization method, which can effectively manipulate output simulation data during the optimization process.
3. The introduction of the AHQGA with two fundamental characteristics to facilitate the limitations of existing research. Firstly, a random coefficient $p(\alpha)$ is applied to preserve the probabilistic manner of quantum measurement, with a view to ensuring the algorithm's performance. Secondly, an auto-adjusting mechanism for the rotation angles to update the population is presented for the sake of convergence speed and globally optimal solutions.
4. The experimental results confirm the effectiveness of the proposed AHQGA algorithm compared to that of HQGA with fixed rotation angles and traditional GA.

The remainder of this paper is organized as follows. Sections 2 and 3 clarify essential quantum computing theories and the procedure for AHQGA, respectively. Subsequently, section 4 illustrates the optimization of the AHQGA for analog circuit sizing while section 5 analyzes the case study of two-stage Miller-compensated op-amp. Simulation results and discussion are presented in section 6, followed by the conclusion of the paper in the last section.

2. Basics of Quantum Computing for AHQGA

Although there are many concepts and principles in the field of quantum computing, we restrict our discussion on quantum fundamentals to those which are essential for the AHQGA. These include essential quantum characteristics and operation quantum gates only.

2.1. Quantum states

Classical computing makes use of bits as the basic unit of information, which has a state of either 0 or 1. On the other hand, quantum computing utilizes the quantum bit (i.e., qubit), which exist as a quantum superposition of these classical bits. Upon measuring the qubit, it will collapse to either the $|0\rangle$ and $|1\rangle$ state [17].

The state of a general qubit $|\psi\rangle$ can be any linear superposition state of $|0\rangle$ and $|1\rangle$ and can be expressed as

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{2}$$

where $|\alpha|^2 + |\beta|^2 = 1$, $|\alpha|^2$ and $|\beta|^2$ are the probabilities that $|\psi\rangle$ collapses to $|0\rangle$ and $|1\rangle$, respectively. $|\psi\rangle$ can also be written in vector form

$$|\psi\rangle = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \tag{3}$$

where $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

2.2. The Bloch sphere

In the Bloch spherical coordinates, the above qubit state $|\psi\rangle$ can be defined as

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle, \tag{4}$$

where $\cos\left(\frac{\theta}{2}\right), e^{i\phi}\sin\left(\frac{\theta}{2}\right)$ are complex numbers representing probability amplitudes of the qubit, and θ and ϕ are the polar angle and azimuthal angle of a spherical coordinate, respectively. Each real value of θ and ϕ corresponds to a three-dimensional point P of a particular qubit on the Bloch sphere [17], as illustrated in Figure 1.

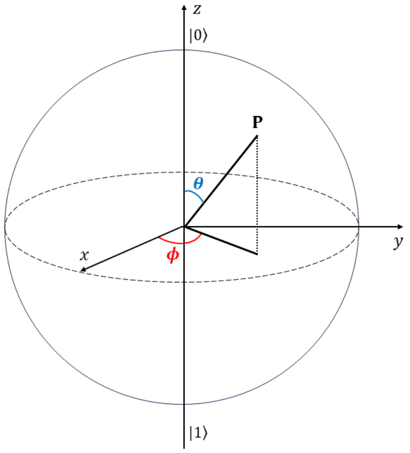


Figure 1: **Representation of a single qubit on the Bloch sphere.**

2.3. Quantum gates

A quantum gate is a gate that performs a unitary operation on a given qubit. To obtain the new quantum state after applying such a transformation, the matrix representation of the quantum gate is multiplied by the initial qubit vector. There are various quantum gates implemented in quantum computing, however, below we focus on introducing the three fundamental quantum gates utilized by AHQGA; the Hadamard, rotation, and Pauli X gate.

2.3.1. Hadamard gate

The Hadamard gate is a 1-qubit gate used to create a superposition of the $|0\rangle$ and $|1\rangle$ basis states. It is defined by the matrix

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{5}$$

2.3.2. Rotation gate

Applying the rotation gate on a given qubit will rotate the qubit around the Bloch sphere to a new state (see Fig. 1). The rotation gate depends on θ and depending on the axial rotation required, will be defined as

$$\begin{aligned} R_x(\theta) &= \begin{pmatrix} \cos(\frac{\theta}{2}) & -i\sin(\frac{\theta}{2}) \\ -i\sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix}, \\ R_y(\theta) &= \begin{pmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix}, \\ R_z(\theta) &= \begin{pmatrix} e^{\frac{-i\theta}{2}} & 0 \\ 0 & e^{\frac{i\theta}{2}} \end{pmatrix}, \end{aligned} \tag{6}$$

where $R_x(\theta)$, $R_y(\theta)$, and $R_z(\theta)$ correspond to rotations around the x-, y-, and z-axis.

2.3.3. Pauli X gate

For the standard basis $|0\rangle$ and $|1\rangle$, the Pauli X gate performs the Boolean NOT operation that maps $|0\rangle \to |1\rangle$ and $|1\rangle \to |0\rangle$. The Pauli X gate is defined by

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{7}$$

3. AHQGA

The HQGA is a cross-disciplinary algorithm that incorporates principles of quantum computing and features of classical GA. Since qubits offer higher capability of computation and speed, classical information is encoded into qubits in HQGA to exploit that advantage. In our improved version of HQGA, AHQGA, we utilize an auto-adjusting scheme for rotation angles to improve the convergence speed and avoid premature convergence. The AHQGA flowchart is presented in Figure 2, with subsequent elaboration of each implementation step in the following subsections.

The optimization framework takes the setup parameters for the population as well as the algorithm as its inputs. Furthermore, the stop condition is chosen as the maximum number of iterations for each optimization process. When the stop condition is satisfied, optimal values for the design variables are produced as the output.

3.1. Quantum Population Initialization

A quantum population consists of many quantum chromosomes. The i^{th} chromosome, or individual, is defined by an n-qubit sequence as

$$|\psi\rangle^i = \sum_j c_i |\psi_j\rangle = \begin{pmatrix} \alpha_1 & \alpha_2 & ... & \alpha_j & ... & \alpha_n \\ \beta_1 & \beta_2 & ... & \beta_j & ... & \beta_n \end{pmatrix}_i \tag{8}$$

where $|\psi_j\rangle = \begin{pmatrix} \alpha_j \\ \beta_j \end{pmatrix}$ is the j^{th} gene.

To initialize a quantum population, the probability amplitude of each gene in the chromosomes must first be equalized. This can be achieved by applying the Hadamard gate to an array of $|0\rangle$. Each gene in the chromosomes is thus expressed as

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \tag{9}$$

Next, rotation is performed (see (6)) with a random angle $\left(\theta \in (0; \frac{\pi}{2}]\right)$ for each gene. Upon completion, the j^{th} gene of the i^{th} chromosome in the quantum population is given by

$$\begin{pmatrix} \alpha_{ji} \\ \beta_{ji} \end{pmatrix} = \begin{pmatrix} \cos(\theta_{ji}) & -\sin(\theta_{ji}) \\ \sin(\theta_{ji}) & \cos(\theta_{ji}) \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \tag{10}$$

3.2. Measurement

The measurement in the standard basis is expressed by the collapse of the wave function as

$$\begin{cases} x_{ji} = 0, \text{ if } p(\alpha) \leq \left|\alpha_{ji}\right|^2, \\ x_{ji} = 1, \text{ otherwise,} \end{cases} \tag{11}$$

where $p(\alpha)$ is a random number in the range $[0;1)$ and x_{ji} is the j^{th} gene of the i^{th} chromosome in the classical population. To maintain the probabilistic characteristics of quantum observation, it is necessary that $p(\alpha)$ not be fixed to a constant value as implemented in [11, 12].

Therefore, after performing measurements on the whole quantum population, the classical population is obtained with the vector representation as

$$(x_{1i} \quad x_{2i} \quad ... \quad x_{ni}), \quad i = \overline{1, m},$$

which contains m chromosome vectors, and each chromosome vector consists of n genes.

3.3. Decoding

Each binary bit sequence $\left(x_{1i} \quad ... \quad x_{ji} \quad ... \quad x_{ni}\right)_i$ corresponding to a typical chromosome is obtained after performing a measurement in the standard basis. They can then be decoded into a decimal number to represent a value of the optimization variables.

The formula for the decoding step is given by

$$value = \text{LB} + \frac{decimal}{2^n} \times (\text{UB} - \text{LB}), \tag{12}$$

where *value* is the real number of the optimization variable, *decimal* is the unsigned decimal value of the binary sequence $\left(x_{1i} \quad ... \quad x_{ji} \quad ... \quad x_{ni}\right)_i$, and LB and UB are the lower and upper bounds of the optimization variables, respectively.

3.4. Fitness Evaluation

The step of fitness evaluation aims to find the fitness values of all individuals in the population based on a predetermined fitness function (or objective function). Depending on the application or the performance metrics to be optimized, the fitness function can be selected differently. For

Table 1: **Lookup table for the rotation angle based on the current chromosome.**

$\boldsymbol{current_j}$	$\boldsymbol{best_j}$	$\boldsymbol{f(current_j) \geq f(best_j)}$	$\boldsymbol{\triangle\theta_j}$
0	0	False	0
0	0	True	0
0	1	False	$+\triangle\theta$
0	1	True	0
1	0	False	$-\triangle\theta$
1	0	True	0
1	1	False	0
1	1	True	0

any application, the fitness function plays a crucial role as the criterion to evaluate the "fitness" or quality of potential solutions, individuals or chromosomes in specific, within the population. As a result, the evolutionary process can be trained in a fitness-oriented manner. In Section 4, the choice of objective function for the multi-objective optimization of the two-stage Miller-compensated op-amp will be explicated.

3.5. Quantum Rotation

Quantum rotation utilizes the transformation matrices in (6) of the rotation gate. The purpose of it is to approximate the state of the currently considered chromosome "*current*" to that of the currently best chromosome over the completed iterations "*best*".

As illustrated in Figure 3, this gate either increases or decreases the probability amplitude of genes towards the chromosome with the best fitness. Hence, the update of the j^{th} gene of an arbitrary chromosome is written as

$$\begin{pmatrix} \alpha_j^{t+1} \\ \beta_j^{t+1} \end{pmatrix} = \begin{pmatrix} \cos(\triangle\theta_j) & -\sin(\triangle\theta_j) \\ \sin(\triangle\theta_j) & \cos(\triangle\theta_j) \end{pmatrix} \begin{pmatrix} \alpha_j^{t} \\ \beta_j^{t} \end{pmatrix}, \tag{13}$$

where $\begin{pmatrix} \alpha_j^{t} \\ \beta_j^{t} \end{pmatrix}$ and $\begin{pmatrix} \alpha_j^{t+1} \\ \beta_j^{t+1} \end{pmatrix}$ are the quantum states before and after updating.

Table 1 is a look-up table that shows the rotation scheme of each gene of the current chromosome with respect to that of the best chromosome. From the table, the $\triangle\theta$ value directly affects the evolution rate of the population. If $\triangle\theta$ is too large, the algorithm may converge to a local optimal solution; if $\triangle\theta$ is too small, the convergence rate of the algorithm will be slow [10].

In other words, $\triangle\theta$ influences the convergence speed of the algorithm. If $\triangle\theta$ is a constant value throughout the algorithm, depending on the optimization problem, a suitable value for $\triangle\theta$ is essential, which might be a time-consuming trial-and-error process. Therefore, instead of fixing $\triangle\theta$ as done previously [13, 14], an adaptive strategy for the rotation angle is applied to prevent premature convergence and improve the convergence speed. This adjusting mechanism of the AHQGA is given by [10], where the adaptive rotation angle is calculated as

$$\triangle\theta = \left[1 - \frac{f(current_j)}{f(best_j)}\right] \triangle\theta_0 \tag{14}$$

where $\triangle\theta_0$ is the fixed coefficient of rotation angle.

3.6. Selection

The selection step plays an essential role in our AHQGA's implementation by determining the composition of the next generation based on the fitness of individuals in the current population. Based on the concept of natural selection, this process ensures the preferential reproduction of individuals with higher fitness, with a view to reproducing promising solutions as well as improvements over generations. Our research employs the tournament selection mechanism to take full advantage of the above-mentioned merits, which improves the algorithm in comparison to previous works where the selection process was not used [12, 14, 13]. This implementation is expected to contribute to the efficiency of AHQGA in terms of finding optimal or near-optimal solutions.

3.7. Quantum Crossover

The quantum crossover operator involves the permutation between two randomly chosen qubits, with a specific crossover rate. Two random quantum chromosomes are considered before measurement, which can be expressed as

$$\begin{pmatrix} \alpha_{1k} & ... & \alpha_{jk} & ... & \alpha_{nk} \\ \beta_{1k} & ... & \beta_{jk} & ... & \beta_{nk} \end{pmatrix}_k, \begin{pmatrix} \alpha_{1t} & ... & \alpha_{jt} & ... & \alpha_{nt} \\ \beta_{1t} & ... & \beta_{jt} & ... & \beta_{nt} \end{pmatrix}_t.$$

If the crossover probability is satisfied, two new quantum individuals can be obtained as

$$\begin{pmatrix} \alpha_{1k} & ... & \alpha_{jt} & ... & \alpha_{nt} \\ \beta_{1k} & ... & \beta_{jt} & ... & \beta_{nt} \end{pmatrix}_k, \begin{pmatrix} \alpha_{1t} & ... & \alpha_{jk} & ... & \alpha_{nk} \\ \beta_{1t} & ... & \beta_{jk} & ... & \beta_{nk} \end{pmatrix}_t,$$

where quantum crossover was performed at the j^{th} gene.

3.8. Quantum Mutation

Quantum mutation is performed using the the Pauli X gate (see (7)) with a certain mutation rate, which is a setup parameter representing the likelihood that a chromosome would undergo mutation process (usually in the range of [0.001;0.01]). If the mutation probability falls in the criteria range, the result of mutating the j^{th} gene of the i^{th} chromosome produces the quantum state of the gene after mutation which is written as

$$\begin{pmatrix} \alpha_{ji}^{*} \\ \beta_{ji}^{*} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha_{ji} \\ \beta_{ji} \end{pmatrix} = \begin{pmatrix} \beta_{ji} \\ \alpha_{ji} \end{pmatrix}. \tag{15}$$

From (15), it is clear that quantum mutation step swaps the probability amplitudes of the mutated qubits.

4. Optimization Problem Formulation

This section demonstrates the essential components to formulate the analog sizing optimization problem, including its procedure, optimization framework, and mathematical generalization.

4.1. Procedure for Analog Circuit Sizing Optimization

The optimization process of analog circuit sizing in general is described in the following paragraphs. Although these steps can be implemented interchangeably, the presented order is recommended for the purpose of convenience and clear orientation.

4.1.1. Topology Selection

Each analog circuit block has various options for its circuit topology. Depending on the specific technical requirements to optimize one or more design parameters and the constraints of the parameters, appropriate circuit configuration should be considered thoroughly. This helps leverage the strength and mitigate the limitations of the chosen circuit topology to meet the predetermined technical requirements.

In short, although the initial stage of determining the optimal circuit topology may require a significant amount of time, it helps circuit designers obtain a clear direction to make the most suitable decisions.

4.1.2. Parameter Selection for Optimization

Depending on the given specific technical requirements, analog circuit parameters for optimization are then determined. Along with that, it also requires identifying the design constraints of each parameter to ensure that the analog circuit optimization process does not prioritize the optimized parameters at the expense of others. Therefore, we aim for the maximum balance between all parameters in the circuit.

In other words, the optimized parameters would be enhanced to the fullest possible extent while the remaining parameters are simultaneously ensured to satisfy the design constraints.

4.1.3. Optimization Design Variables

To formulate an optimization problem for analog circuit sizing, the formulas based on these parameters, such as the dimensions of the MOSFET channel $\big($i.e., length (L) and width $(W)\big)$, resistance, and capacitance, need to be established. Next, besides the parameters that need to be held constant, it is required to determine the variables that directly affect the design parameters to select appropriate optimization variables.

Since each technology node and corresponding device has different ranges of values for the parameters mentioned above, their limits must be set. Furthermore, to possibly approach global optimum values with fewer iterations, it is crucial to limit the range of values of the optimization variables to narrow the search space, which increases the probability of improving the efficiency of the optimization process.

4.1.4. Objective Function and Design Constraints

Determining the objective function for the optimization problem of a similar circuit not only clarifies the purpose of the optimization problem but also serves as a crucial condition for implementing the AHQGA. The reason is that we need a specific value from the objective function to quantitatively evaluate a particular set of parameters for a similar circuit, thereby identifying the better set of parameters for the circuit. Furthermore, as mentioned above, to avoid imbalance among the parameters in the circuit, additional constraints on the parameters exclusive from the group of optimized parameters need to be imposed to achieve the maximum balance between those parameters.

4.2. AHQGA-Spectre Framework

For the multi-objective optimization of the two-stage Miller-compensated op-amp, it is necessary that the structure of the AHQGA described in Figure 2 be modified at the "Fitness Evaluation" step. For the aforementioned analog sizing optimization problem, the "Fitness Evaluation" block should include two substeps, namely "Circuit Simulation" and "Fitness Calculation". The first substep is the simulation of the required design parameters, implemented by simulation software such as the Spectre simulator of Cadence Virtuoso tool. The other substep is the calculation of fitness values based on simulation data of the previous step, which is performed by a Python-based script.

Furthermore, the choice of platform for both the optimization core and circuit simulation must be clarified. In this research, the AHQGA is implemented using Python programming language since Python incorporates a variety of built-in libraries for quantum and optimization algorithms. Furthermore, Spectre is chosen as the circuit simulator in our platform instead of HSPICE ([6], [15]), since simulation output data from HSPICE are mostly presented in a predetermined format, requiring an additional step of using scripting languages to acquire compatible output format for AHQGA's processing. Meanwhile, Ocean-based scripts of the Spectre simulator can integrate SKILL's programming syntax to directly arrange simulation results according to users' preference [2]. By virtue of the circuit simulator's role in the entire optimization system and the convenience of output data format, Spectre is the preferred practical choice.

The connection between Spectre and Python is demonstrated as follows. Initially, Python generates values for the design variables' population, which are then transmitted to Spectre through an Ocean-based script. This Ocean script handles automated circuit simulations using the provided values, and the results are returned to Python for the fitness evaluation step of the chromosomes. This iterative process continues until the stop condition of the algorithm. The Python-Spectre interaction is illustrated in greater detail by the block diagram presented in Figure 4.

4.3. Generalization of the Analog Circuit Sizing Optimization Problem

In general, an optimization problem of analog circuit sizing can be formulated as

$$\begin{aligned} &\mathit{Optimize}\ \mathrm{F}(\vec{x}) \\ &s.t.\ \Omega = \{\vec{x} \in \Re^n \mid \mathrm{G}(\vec{x}) \leq 0\} \end{aligned} \tag{16}$$

where $\vec{x}$ is a multi-dimensional vector of n optimization variables in $\Re^n$, delimited by the lower bound x^i_{min} and upper bound x^i_{max}, which are given by $x^i_{min} \leq x_i \leq x^i_{max}$ $(i \in \{1, 2, ..., n\})$. $F(\vec{x})$ represents the vector of m objectives $f_1(\vec{x}), f_2(\vec{x}), ..., f_m(\vec{x})$ to be minimized or maximized, where $m = 1$ and $m > 1$ corresponds to single-objective and multi-objective optimization problem, respectively [18]. $G(\vec{x})$ describes the vector of constraints that must be satisfied to guarantee the feasibility of the solution, which might include circuit setup conditions, saturation conditions, and constraints for design parameters.

5. Case study: The two-stage Miller-compensated operational amplifier

Operational amplifiers (op-amps), with the primary aim of amplifying voltage signals and providing a stable output, are considered one of the fundamental blocks in analog circuit design. They are integrated as sub-blocks in several different larger blocks with higher levels of complexity such as bandgap references, low-dropout regulators, and analog-to-digital converters. Depending on the applications and design specifications, the role of the op-amp differentiates; therefore, some performance metrics are prioritized at the dispense of others [15]. With a view to designing an op-amp that can be virtually applied to different purposes, a balance among its universal design parameters is necessary while simultaneously keeping other parameters at acceptable levels. This corresponds to the purpose of the multi-objective optimization, which should be the primary motivation to select the op-amp as a case study for the multi-objective optimization of analog circuit sizing.

As shown in Figure 5, the two-stage Miller-compensated op-amp consists of two stages, namely an operational transconductance amplifier (OTA) stage cascading with a common source (CS) stage.

In this work, we focus on the 65 nm technology node, with the supply voltage V_{DD} ranging from 1 V to 1.2 V. The maximum achievable voltage gain for a single-stage amplifier is approximately 28 dB, which is much less than the required gain for any application [19]. Therefore, techniques to increase gain are necessary. Conventional cascode op-amps can exhibit a much higher gain at around 80 dB; however, due to stacked transistors between the power supply V_{DD} and ground, MOSFET's saturation condition with acceptable margin might not be ensured. Hence, they should be considered unsuitable for low voltage applications [20].

The aforementioned issue can be facilitated by cascading several amplifier stages to form a multi-stage amplifier [20]. Regarding the first stage, since the OTA produces a single-ended output from differentially paired inputs and exhibits a high voltage gain, it is chosen for the convenience of cascading the following stages. However, cascading introduces high-impedance nodes and hence leads to close placement of several poles, affecting the op-amp's stability. Although a large capacitor can be used to move the dominant pole frequency to a lower frequency and a high bias current can also be used to push the non-dominant pole to a higher value, these solutions are not practical. As a result, there arises the need for frequency compensation techniques, in which Miller compensation is presented in detail below.

It is also worth noticing some key points when designing the two-stage Miller-compensated op-amp. According to Figure 5, assume that the current flowing through the OTA stage is I_{REF} and the current flowing through the CS stage is kI_{REF}. This results in

$$\frac{W_5}{L_5} = k\frac{W_4}{L_4} = k\frac{W_7}{L_7} \tag{17}$$

where W_i, L_i are the channel width and length of MOSFET M_i, respectively, $L_5 = L_4 = L_7$, and k is a constant.

Also, the condition to ensure the symmetry of the OTA stage is expressed as

$$\frac{W_0}{L_0} = \frac{W_1}{L_1}, \frac{W_2}{L_2} = \frac{W_3}{L_3}. \tag{18}$$

Furthermore, for the op-amp to work properly, all MOSFETs need to be biased to saturate. Moreover, margins of 30 mV for the gate-source voltage (V_{GS}) as well as overdrive voltage (V_{OV}) should also be ensured in this work, for possible applications later on.

5.1. The Need for Miller Compensation

Figure 6 demonstrates a two-stage op-amp with only two poles at nodes $V_{out_{OTA}}$ and V_{out} (other poles are negligible). From the circuit, it can be inferred that $R_{OTA} \approx r_{o1} \parallel r_{o2}$ and $C_{OTA} = C_{DB1} + C_{DB2} + C_{GB6}$ are the respective output resistance and capacitance of the first stage, while $R_{CS} = r_{o5} \parallel r_{o6}$ and $C_{CS} = C_{DB5} + C_{DB6}$ are the respective output resistance and capacitance of the second stage (CS). By approximating poles with nodes as in [4], two poles are obtained as

$$p_1 = \frac{1}{R_{OTA}C_{OTA}},\ p_2 = \frac{1}{R_{CS}C_{CS}}. \tag{19}$$

With this two-pole system, it cannot be completely certain that the phase margin of the op-amp is greater than 60°, which directly affects its stability [4]. Transforming it into a single-pole system (one pole dominates the other) can be achieved by increasing the output resistance or capacitance of either stage. Nevertheless, this increase affects the DC gain and bandwidth of the op-amp. Therefore, adding a Miller compensation capacitor C_C to connect the two nodes $V_{out_{OTA}}$ and V_{out} can carry out the transformation more efficiently (Figure 7).

This compensation capacitor can be further separated by applying the Miller theorem (as shown in Figure 8). The two poles can then be expressed as

$$p_1 = \frac{1}{R_{OTA}[C_{OTA} + C_C(1 + g_{m6}R_{CS})]}, \tag{20}$$

$$p_2 = \frac{1}{R_{CS}[C_{CS} + C_C(1 + \frac{1}{g_{m6}R_{CS}})]}. \tag{21}$$

From (20) and (21), p_1 can be the dominant pole if $p_1 \ll p_2$ with proper $A_{V_{CS}}$. The op-amp can then be considered a one-pole system with better stability.

5.2. Design Parameters

Before analyzing the design parameters of the two-stage Miller-compensated op-amp, its transfer function needs to be determined using exact analysis from the small signal model (as shown in Figure 9).

From the circuit in Figure 9, applying Kirchhoff's Current Law (KCL) at node $V_{out_{OTA}}$ in the frequency domain results in

$$\frac{V_{out_{OTA}}}{R_{OTA}} + \frac{V_{out_{OTA}}}{(sC_{OTA})^{-1}} + g_{m1}V_{in} + \frac{V_{out_{OTA}} - V_{out}}{(sC_C)^{-1}} = 0. \tag{22}$$

Based on (22), $V_{out_{OTA}}$ can be expressed as

$$V_{out_{OTA}} = \frac{V_{out}sC_CR_{OTA} - g_{m1}R_{OTA}V_{in}}{1 + sR_{OTA}(C_{OTA} + C_C)}. \tag{23}$$

Moreover, applying KCL again, but at node V_{OUT} results in

$$\frac{V_{out}}{R_{CS}} + \frac{V_{out}}{(sC_{CS})^{-1}} + g_{m6}V_{out_{OTA}} + \frac{V_{out} - V_{out_{OTA}}}{(sC_C)^{-1}} = 0. \tag{24}$$

Hence, it can be inferred that

$$sV_{out_{OTA}}(C_C - g_{m6}) = V_{out} \times \Big(s\big(C_C + C_{CS}\big) + \big(R_{CS}\big)^{-1}\Big). \tag{25}$$

From (23) and (25), the transfer function is obtained as

$$H(s) = \frac{V_{out}}{V_{in}} = \frac{g_{m1}R_{OTA}g_{m6}R_{CS}(1 - \frac{sC_C}{g_{m6}})}{H_1s^2 + H_2s + 1}, \tag{26}$$

where

$$\begin{aligned} H_1 &= R_{OTA}R_{CS}\Big(C_{OTA}C_{CS} + C_{OTA} \times C_C + C_{CS}C_C\Big), \\ H_2 &= R_{CS}\Big(C_C + C_{CS}\Big) + R_{OTA}\Big(C_C + C_{OTA}\Big) + C_Cg_{m6}R_{OTA}R_{CS}. \end{aligned} \tag{27}$$

Considering the op-amp as a two-pole system, its standard transfer function is given by

$$H(s) = \frac{A_V(1 - \dfrac{s}{z})}{(1 + \dfrac{s}{p_1})(1 + \dfrac{s}{p_2})} = \frac{A_V(1 - \dfrac{s}{z})}{\dfrac{1}{p_1p_2}s^2 + (\dfrac{1}{p_1} + \dfrac{1}{p_2})s + 1}, \tag{28}$$

where A_V is the voltage gain of the op-amp, p_1, p_2 are the two pole frequencies and z is the zero frequency of the op-amp.

When p_1 is the dominant pole ($p_1 \ll p_2$), it results in

$$\frac{1}{p_1} + \frac{1}{p_2} \approx \frac{1}{p_1}, \tag{29}$$

Using this approximation of (29), (28) can be simplified as

$$H(s) = \frac{A_V(1 - \dfrac{s}{z})}{\dfrac{1}{p_1p_2}s^2 + \dfrac{1}{p_1}s + 1}. \tag{30}$$

From (26) and (30), p_1 is defined as

$$p_1 = \Bigg(R_{CS}\big(C_C + C_{CS}\big) + R_{OTA}\big(C_C + C_{OTA}\big) + g_{m6} R_{OTA} R_{CS} C_C \Bigg)^{-1}, \quad (31)$$

which is approximated as

$$p_1 \approx \frac{1}{g_{m6} R_{OTA} R_{CS} C_C}, \quad (32)$$

while $p_1 p_2$ and z take the form

$$p_1 p_2 = \Bigg(R_{OTA} R_{CS}\big(C_{OTA} C_{CS} + C_{OTA} \times C_C + C_{CS} C_C\big) \Bigg), \quad (33)$$

and

$$z = \frac{g_{m6}}{C_C}. \quad (34)$$

From (31)-(33), p_2 can then be defined as

$$p_2 = \frac{g_{m6} C_C}{C_{CS} C_{OTA} + C_C C_{OTA} + C_C C_{CS}} \approx \frac{g_{m6}}{C_{CS}} \approx \frac{g_{m6}}{C_L}. \quad (35)$$

Having derived all the necessary expressions for the poles p_1, p_2 and z from the op-amp transfer function $H(s)$, the main design parameters of the op-amp for optimization is summarized in the following subsections.

5.2.1. Small-signal Voltage Gain

From (26) and (30), the small-signal voltage gain, which is the product of the voltage gain of the two stages (CS stage and OTA stage), is expressed as

$$A_V = A_{V_{OTA}} A_{V_{CS}} = g_{m1}(r_{o1} \parallel r_{o2}) g_{m6}(r_{o5} \parallel r_{o6}). \quad (36)$$

where $A_{V_{OTA}}$ and $A_{V_{CS}}$ are small-signal voltage gains of the OTA and CS stages, respectively.

5.2.2. Common Mode Rejection Ratio (CMRR)

As the second CS stage deals with single-ended signals, the common mode rejection ratio (CMRR) of the two-stage Miller-compensated op-amp can be equivalent to the CMRR of the first OTA stage with differential signals. Referring to [4], CMRR can be calculated as

$$\text{CMRR} = \frac{A_{V_{OTA}}}{A_{V_{OTA-CM}}} = g_{m2}(1 + 2g_{m1}r_{o4})(r_{o1} \parallel r_{o2}) \tag{37}$$

where $A_{V_{OTA-CM}}$ is the common-mode voltage gain of the op-amp.

5.2.3. Unity Gain Bandwidth

The unity gain bandwidth is obtained as

$$\text{UGB} = A_V \times \text{BW} \approx A_V \times p_1 = \frac{g_{m1}}{C_C} \tag{38}$$

where UGB and BW are the unity gain bandwidth and bandwidth of the op-amp, respectively.

5.2.4. Phase Margin

The phase margin can be expressed as

$$\text{PM} = \pm 180 - tan^{-1}\left(\frac{\text{UGB}}{p_1}\right) - tan^{-1}\left(\frac{\text{UGB}}{p_2}\right) - tan^{-1}\left(\frac{\text{UGB}}{z}\right). \tag{39}$$

For a stable op-amp, a PM greater than 60° is preferable [4].

5.2.5. Power Dissipation

The power dissipation of the op-amp equals the sum of power consumption of the OTA and CS stage, which is given by

$$\text{Power} = V_{DD}I_{total} = V_{DD}(I_4 + I_5), \tag{40}$$

where $I_{total} = I_4 + I_5$ is the total current of the op-amp, and I_4 and I_5 are the currents flowing through MOSFETs M_4 and M_5, respectively.

5.2.6. Slew Rate

The slew rate of the op-amp, which is the rate of change of the output voltage with respect to time, can be expressed as

$$\text{SR} = \frac{I_4}{C_C}. \tag{41}$$

5.3. Choice of the Objective Function

As presented in Section 3, an objective function (fitness function) is considered a standard to assess the individuals of the population and thereby exerts a direct influence on the evolutionary process. Therefore, the choice of a suitable objective function is crucial.

In the context of analog sizing optimization, the Figure of Merit (FoM), which typically refers to a quantitative measure to evaluate the performance of a system, can play the same role as the objective function. The FoM to compare the performance of the op-amp is given in [21] as

$$\text{FoM} = \frac{\text{UGB} \times C_L}{I_{total}}, \tag{42}$$

where UGB is the unity gain-bandwidth product, C_L is the load capacitance at the output node, and I_{total} is the total current of the op-amp.

Although (42) evaluates the efficiency of the op-amp based on its unity gain-bandwidth product as well as the drivability of the load capacitance per unit current, the phase margin of the op-amp is overlooked. An op-amp with a larger value for the FoM of (42) is considered with better quality; however, an increase in C_L leads to a decrease in p_2 as in (35), at the expense of the op-amp's phase margin and hence stability. In other words, considering (42), a large FoM value is no longer meaningful if the phase margin is too small [16], indicating the unsuitability of the above-mentioned objective function.

Taking into account the role of the op-amp's phase margin, the enhanced version of the FoM, which is applied in this research based on [16], is expressed as

$$\text{FoM} = \frac{\text{UGB} \times C_L}{I_{total}} \times \frac{tan(\text{PM})}{tan(PM_{REF})}, \tag{43}$$

where PM_{REF} is the reference phase margin of the op-amp and chosen with the standard value of 60° [4].

5.4. Initialization Steps for Optimization

Regarding the two-stage Miller-compensated op-amp, its design is executed in the TSMCN65 process. The op-amp's setup condition includes $V_{DD} = 1.2$ V, $I_{REF} = 20$ μA, and $C_L = 1$ pF. The input common-mode voltage for both V_{inn}, V_{inp} is $V_{in_{CM}} = 650$ mV.

According to (17), we declare three optimization variables as W_5, W_{47}, and L_{457}. Moreover, (18) indicates four additional variables: $W_{01}, L_{01}, W_{23}, L_{23}$.

Similarly, W_6, L_6, and C_C are also selected as optimization variables. In total, there are ten variables for our optimization process, namely W_{01}, L_{01}, W_{23}, L_{23}, W_5, W_{47}, L_{457}, W_6, L_6, C_C.

The specifications for the op-amp are as follows. First of all, every MOSFETs should operate in the saturation region with their margins of 30 mV ensured for both the gate-source voltage V_{GS} and the overdrive voltage. For the sake of convenience, the mentioned condition is labeled $cond_1$. When $cond_1$ is satisfied, the standard for the op-amp's performance metrics ($cond_2$) is expressed as

$$A_V > 50\ dB, \tag{44a}$$

$$\text{UGB} > 50\ MHz, \tag{44b}$$

$$\text{PM} > 60^\circ, \tag{44c}$$

$$\text{CMRR} > 50\ dB, \tag{44d}$$

$$\text{Power} < 250\ \mu W, \tag{44e}$$

$$\text{SR} > 50\ V/\mu s. \tag{44f}$$

It should be noticed that $cond_2$ is satisfied only when all design specifications of the op-amp above are met.

As a higher value of the FoM is preferable and $cond_1$ is prioritized over $cond_2$, the FoM formula can be rewritten with the addition of $cond_1$ and $cond_2$ as

$$\text{FoM} = \begin{cases} -1 & \text{if } cond_1 = 0, \\ 0 & \text{if } cond_1 = 1, cond_2 = 0, \\ \frac{\text{UGB}\times C_L}{I_{total}} \times \frac{tan(\text{PM})}{tan(60^\circ)} & \text{otherwise.} \end{cases} \tag{45}$$

where the value of 0 and 1 is equivalent to whether each condition passes or not.

For the cases when both $cond_1$ and $cond_2$ are not satisfied, the values of -1 and 0 are chosen for the FoM with the purpose of excluding the equivalent potential solutions. Moreover, the order of -1 and 0 corresponds to the priority of $cond_1$ and $cond_2$.

So as to ensure the saturation condition to the greatest possible extent, the bounds for optimization variables are determined as follows. Firstly, all MOSFETs' widths and lengths are initialized with the minimum values of the process node. Then, the widths and lengths of one MOSFET or one pair of symmetric MOSFETs are adjusted while others might be kept constant or vary based on the basic knowledge of MOSFET's operation. This step is to find the temporary bounds for the optimization variables. Subsequently, this iterative process continues for the remaining MOSFET or pair of symmetric MOSFETs, in which the temporary bounds might be changed accordingly for better suitability. Finally, the bounds for optimization variables are obtained. Notice that the aforementioned procedure's purpose is to possibly ensure the saturation condition to limit the search space for better efficiency in finding the optimal FoM value.

As a result, the bounds for the declared optimization variables are obtained as

$$\begin{aligned}
&W_{01} \in [0.85\mu m, 4\mu m], L_{01} \in [0.23\mu m, 0.4\mu m],\\
&W_{23} \in [0.7\mu m, 1\mu m], L_{23} \in [0.06\mu m, 0.4\mu m],\\
&W_{47} \in [2\mu m, 2.8\mu m], W_5 \in [18\mu m, 23\mu m],\\
&L_{457} \in [0.1\mu m, 1\mu m], W_6 \in [16\mu m, 22\mu m],\\
&L_6 \in [0.25\mu m, 0.5\mu m], C_C \in [0.3pF, 1pF].
\end{aligned} \tag{46}$$

These bounds are obtained from designer's experience and behave like rules of thumb to ensure control on the optimization process.

In essence, our optimization problem can be summarized as

$$\begin{aligned}
maximize\ & \mathrm{FoM}(W_{01}, L_{01}, W_{23}, L_{23}, W_5, W_{47},\\
& \qquad L_{457}, W_6, L_6, C_C) && \text{(47a)}\\
s.t.\ \ & V_{DD} = 1.2V, && \text{(47b)}\\
& V_{in_{CM}} = 650mV, && \text{(47c)}\\
& I_{REF} = 20\mu A, && \text{(47d)}\\
& C_L = 1pF, && \text{(47e)}\\
& (44), (46). && \text{(47f)}
\end{aligned}$$

6. Numerical Results and Discussion

In terms of AHQGA, the quantum chromosome representation specifies 16 qubits, and the population size is set to 16 individuals. While crossover has a high probability, the mutation probability is typically low. As shown by [22], the rates for crossover (r_{cross}) and mutation (r_{mut}) fall within the ranges $[0.8, 0.95]$ and $[0.001, 0.05]$, respectively.

In our study, simulation results showed that the highest value for the fitness function in (45) was 0.5654 which was achieved with $r_{cross} = 0.8, r_{mut} =$

0.05 and $\triangle\theta_0 = 0.005\pi$. With all specifications satisfied, the fitness value of 0.5654 includes the optimum design parameters, as presented in Table 2.

Table 2: **Optimum design parameters achieved for the two-stage Miller-compensated op-amp with their corresponding specifications.**

Design parameters	**Specification**	**Optimum value**
A_V	$> 50\ dB$	$50.58\ dB$
UGB	$> 50\ MHz$	$61.51\ MHz$
PM	$> 60°$	$68.75°$
CMRR	$> 50\ dB$	$51.73\ dB$
Power	$< 250\ \mu W$	$193.78\ \mu W$
SR	$> 50\ V/\mu s$	$61.20\ V/\mu s$

Moreover, fitness values over 100 iterations of the AHQGA are illustrated in Figure 10. Although the values for various op-amp design parameters may vary unpredictably throughout the iterations, their corresponding fitness values followed a monotonous upward trend as in Figure 10. This conforms to the algorithm's principle of survival to the fittest which produces better individuals for each generation.

Regarding the optimization variables of AHQGA, it is worth taking into account the process grid of 5 nm of the TSMCN65 process. Therefore, the widths and lengths of MOSFETs after the optimization process needed to be rounded to reasonable values. The post-optimization sizes of all MOSFETs in the circuit are presented in Table 3, while Table 4 outlines the performance comparison between the two-stage Miller-compensated op-amp presented in this study and other research.

From Table 4, in comparison to [3], it is evident that our work exhibits surpassing values for FoM as well as A_V, PM, and $CMRR$. Since our A_V is more than twice that of [3] (50.58 dB compared to 21.6 dB), the amount of current in the circuit should be greater, primarily resulting in our higher value of power consumption. Moreover, since our work's C_C, C_L are larger (0.28 pF, 1 pF versus 0.06 pF, 0.2 pF, respectively), better UGB and SR from [3] are reasonable since they are inversely proportional to the mentioned capacitances. Regarding [23], the op-amp of our work achieves far better A_V, UGB, and PM. However, the FoM value is not as that good mainly due to the use of a ten-fold output capacitor C_L of [23]. This requires a larger current through the op-amp to drive the load C_L though, leading to a power

Table 3: **Post-optimization sizing results of optimized two-stage Miller-compensated op-amps.**

Device	**Size (W/L)**
M_0, M_1	$1.47\mu m/0.32\mu m$
M_2, M_3	$0.85\mu m/0.15\mu m$
M_4	$2.45\mu m/0.55\mu m$
M_5	$18.02\mu m/0.55\mu m$
M_6	$16.38\mu m/0.25\mu m$
M_7	$2.45\mu m/0.55\mu m$
C_C	$0.28pF$

Table 4: **Performance summary of the proposed two-stage op-amp versus those of other different research works**

Performance metrics	[**3**]	[**23**]	[**24**]	[**5**]	**This work**
CMOS process (nm)	65	65	180	180	65
Supply voltage (V)	1.1	1.1	1.8	1.8	1.2
C_L (pF)	0.2	10	10	1	1
A_V (dB)	21.6	43	75.62	75.35	50.58
UGB (MHz)	169.7	28	16.78	45.11	61.51
PM (°)	62.4	60	62.48	61.24	68.75
$CMRR$ (dB)	35.5	N.A	N.A	N.A	51.73
Power (μW)	89	321	220	123.156	193.78
SR (V/μs)	288	N.A	15.88	N.A	61.20
FoM (HzF/A)	0.1432	0.9595	1.5214	0.6936	0.5654

Table 5: **Optimal solutions from from proposed AHQGA, HQGA with fixed $\triangle\boldsymbol{\theta}$, and traditional GA**

Criterion	**AHQGA**	**HQGA**	**GA**
Optimal FoM value	0.5654	0.5581	0.5916
Iterations to reach optimal FoM value	49	44	94

consumption virtually twice as much as that of our work, which is not covered in the FoM formula.

In addition, our UGB as well as PM results dominate those of [24] and [5]; our value for UGB is approximately 4 and 1.5 times better, respectively. Meanwhile, the voltage gain A_V in our work is around 25 dB lower, most likely because of the higher range for the transistors' lengths as well as the higher supply voltage for the 180 nm process compared to the 65 nm process. Furthermore, the comparison of [24] with our work regarding the metrics of power consumption bears a strong resemblance to [23], with a larger C_L and corresponding higher power. Additionally, the lower power dissipation value of [5] versus that of our work might be attributed to the abundant room for the 1.8 V supply voltage that does not demand much current through the circuit. In general, although our work surpasses [24] and [5] in the UGB and PM aspects, lower FoM was observed. This results from the difference in technology node (180nm and 65nm), and lower supply voltage (1.2V to 1.8V), which made it harder to design the circuit. Nevertheless, our work still proves its efficiency and feasibility based on other performance metrics.

With the purpose of further assessment of the AHQGA, Figure 11 illustrates comparisons of the proposed FoM, accompanied by that of both HQGA with fixed $\triangle\theta = 0.025\pi$ and GA. In order for the analysis to be judicious, all optimization algorithms were set up with the same initial conditions, including a similar initial population, crossover and mutation rate ($r_{cross} = 0.8$ and $r_{mut} = 0.05$) as well as the number of iterations (100 total). Table 5 provides further deliberations on the optimization results of the three mentioned algorithms.

According to Table 5 and Figure 11, AHQGA dominates its HQGA counterpart for the optimal value of FoM, where AHQGA achieves the 0.5654 value compared to 0.5518 of HQGA. Moreover, escaping from the so-called "FoM - zero" search space, which occurs in solutions that meet saturation conditions but do not satisfy specifications as in (44), should be considered essential as this will prevent the optimization algorithm from being trapped in this undesirable search space. AHQGA was successful in this respect at the 15^{th} iteration, which was two times faster than HQGA at around the 30^{th} iteration. In other words, AHQGA completely prevails over HQGA, corresponding to the purpose of the auto-adjusting scheme as an upgrade for HQGA.

Lastly, it is undeniable that AHQGA is inferior to GA with respect to the optimal FoM value (0.5654 versus 0.5916). However, two key points must be considered in this comparison. First, similar to HQGA, AHQGA exceeded GA concerning the required iterations to get rid of the "FoM - zero" search space by the same two-fold ratio. Second, the number of iterations to reach the optimal solution of AHQGA far outnumbered that of GA, namely 49 versus 94 iterations. In summary, the two above-mentioned benefits of AHQGA over GA might compensate for the slightly lower optimal FoM value.

7. Conclusion

In conclusion, AHQGA, thanks to its best-individual-based rotation mechanism, can be considered as a novel approach to employing quantum-assisted optimization algorithms for analog circuits. With a view to facilitating designers' effort and experience, the AHQGA-Spectre simulation-based platform to optimize the two-stage Miller-compensated op-amp is proposed in this paper. Compared to other published work, results are promising with an optimal FoM value of 0.5654, consisting of $A_V = 50.58$ dB, $UGB = 61.51$ MHz, PM = 68.75°, $CMRR = 51.73$ dB, Power = 193.78 μW and $SR = 61.20$ V/μs. Moreover, for an objective evaluation considering the same setup and initial condition, optimization results implemented by AHQGA were far superior to those obtained by HQGA, while the results from AHQGA can be comparable to that of GA in terms of the trade-off between optimal FoM value and convergence speed. Most significantly, due to the mentioned outcomes and the adaptability of the optimization framework, our research might underscore the potential applications of quantum algorithms in future analog circuit designs, possibly transforming the work approach of analog circuit engineers.

Acknowledgement

We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study.

References

[1] B. Liu, Y. Wang, Z. Yu, L. Liu, M. Li, Z. Wang, J. Lu, F. V. Fernández, Analog circuit optimization system based on hybrid evolutionary algorithms, Integr. VLSI J. 42 (2) (2009) 137–148.

[2] H. Nguyen, T. Hoang, A novel framework of genetic algorithm and spectre to optimize delay and power consumption in designing dynamic comparators, Electronics 12 (16) (2023) 3392.

[3] R. Rashid, N. Nambath, Area optimisation of two stage Miller compensated op-amp in 65 nm using hybrid PSO, IEEE Trans. Circuits Syst. II Exp. Briefs 69 (1) (2022) 199–203.

[4] B. Razavi, Design of Analog CMOS Integrated Circuits, 2nd Edition, McGraw-Hill, New York, 2017.

[5] S. A. Abdelaal, A. Hussein, H. Mostafa, A Bayesian optimization framework for analog circuits optimization, in: Proc. 15th Int. Conf. Comput. Eng. Syst. (ICCES), 2020, pp. 1–4.

[6] S. R. Chowdhury, S. Bhardwaj, J. Kitchen, Design automation of CMOS op-amps using statistical geometric programming, in: Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), 2022, pp. 1575–1579.

[7] A. F. Budak, P. Bhansali, B. Liu, N. Sun, D. Z. Pan, C. V. Kashyap, DNN-Opt: An RL inspired optimization for analog circuit sizing using deep neural networks, in: Proc. 58th ACM/IEEE Design Automat. Conf. (DAC), San Francisco, CA, USA, 2021, p. 1219–1224.

[8] A. Lberni, A. Sallem, M. Alami Marktani, A. Ahaitouf, N. Masmoudi, A. Ali, An efficient multi-objective simulation-based approach for low voltage low power analog ICs, in: Proc. 2020 IEEE International Conference on Design and Test of Integrated Micro and Nano-Systems (DTS), Hammamet, Tunisia, 2020, pp. 1–5.

[9] L.-x. Sha, Y.-y. He, A novel self-adaptive quantum genetic algorithm, in: Proc. 2012 8th International Conference on Natural Computation, Chongqing, China, 2012, pp. 618–621.

[10] Z. Liu, X. Li, J. Jiang, S. Wang, A novel improved quantum genetic algorithm for robot coalition problem, in: Proc. IEEE International Conference on Information and Automation (ICIA), Ningbo, China, 2016, pp. 2061–2064.

[11] Y. Sun, M. Ding, Quantum genetic algorithm for mobile robot path planning, in: Proc. 2010 Fourth International Conference on Genetic and Evolutionary Computing, Shenzhen, China, 2010, pp. 206–209.

[12] R. Lahoz-Beltra, Quantum genetic algorithms for computer scientists, Computers 5 (4) (2016) 24.

[13] D. Li, J. Zhao, H. Zhang, P. Qiao, J. Zhuang, An improved many worlds quantum genetic algorithm, in: Proc. 11th International Conference on Natural Computation (ICNC), Zhangjiajie, China, 2015, pp. 210–214.

[14] K.-H. Han, J.-H. Kim, Quantum-inspired evolutionary algorithm for a class of combinatorial optimization, IEEE Trans. Evol. Comput. 6 (6) (2002) 580–593.

[15] H. M.V., B. Harish, An integrated MaxFit genetic algorithm-SPICE framework for 2-stage op-amp design automation, in: Proc. IEEE Comput. Soc. Ann. Symp. VLSI (ISVLSI), Hong Kong, China, 2018, pp. 170–174.

[16] H. Yoshizawa, An improved figure-of-merit equation for op-amp evaluation, IEICE Electron. Exp. 12 (Aug. 2015).

[17] M. A. Nielsen, I. L. Chuang, Quantum Computation and Quantum Information: 10th Anniversary Edition, Cambridge University Press, 2010.

[18] M. Barros, J. Guilherme, N. Horta, Analog Circuits and Systems Optimization based on Evolutionary Computation Techniques, Springer, 2010.

[19] R. Nagulapalli, K. Hayatleh, S. Barker, S. Zourob, N. Yassine, B. N. K. Reddy, A technique to reduce the capacitor size in two stage Miller compensated op-amp, in: Proc. 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Bengaluru, India, 2018, pp. 1–4.

[20] R. Nagulapalli, K. Hayatleh, S. Barker, B. N. K. Reddy, B. Seetharamulu, A low power Miller compensation technique for two stage op-amp in 65nm CMOS technology, in: Proc. Int. Conf. Comput. Commun. Netw. Technol. (ICCCNT), Kanpur, India, 2019, pp. 1–5.

[21] W. Sansen, Analog Design Essentials, Springer, 2006.

[22] S. Chien, X.-S. Yang, T. Ting, Bio-Inspired Computation and Optimization: An Overview, Elsevier, 2015.

[23] M. El-Sisi, M. Dessouky, Procedural analog design automation using building block optimization, in: Proc. IEEE International Conference on Electronics, Circuits, and Systems (ICECS), Cairo, Egypt, 2015, pp. 474–477.

[24] P. Das, B. Jajodia, Design automation of two-stage operational amplifier using multi-objective genetic algorithm and SPICE framework, in: Proc. Int. Conf. Invent. Comput. Technol. (ICICT), Nepal, 2022, pp. 166–170.

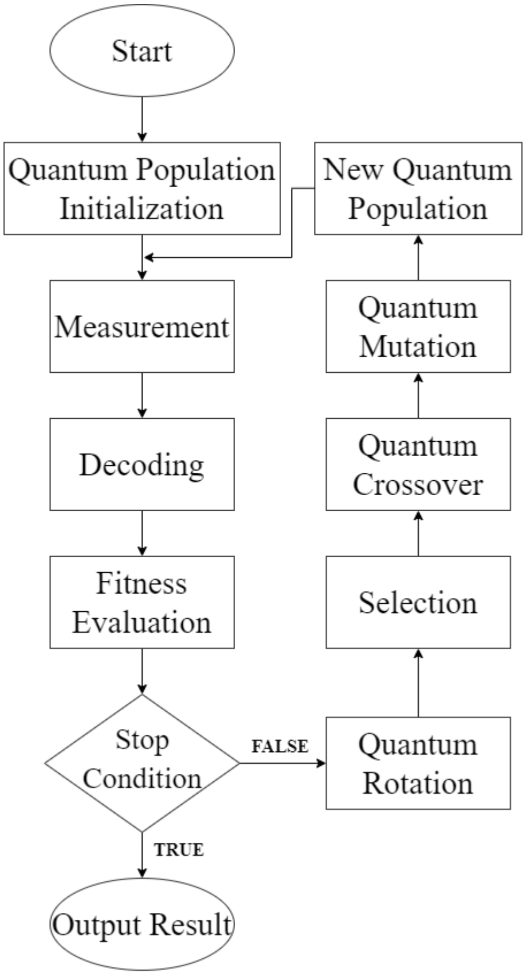


Figure 2: **Flowchart of AHQGA.**

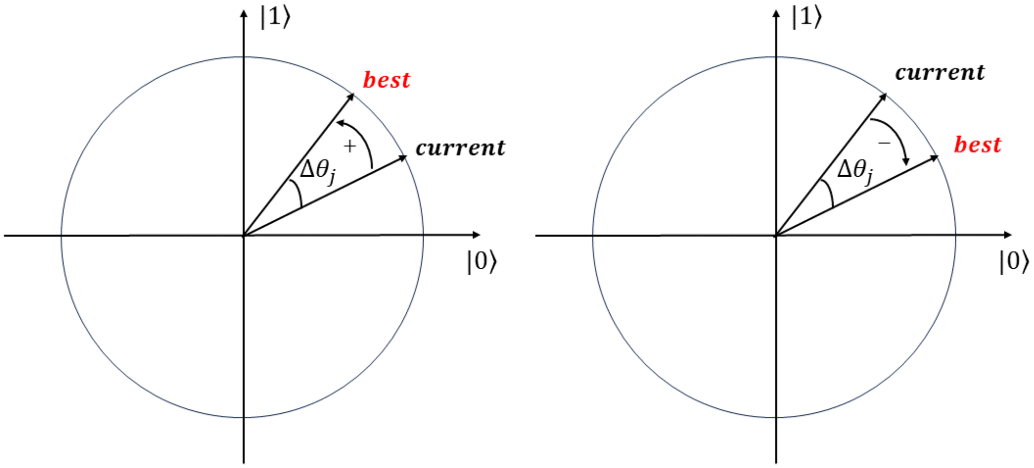
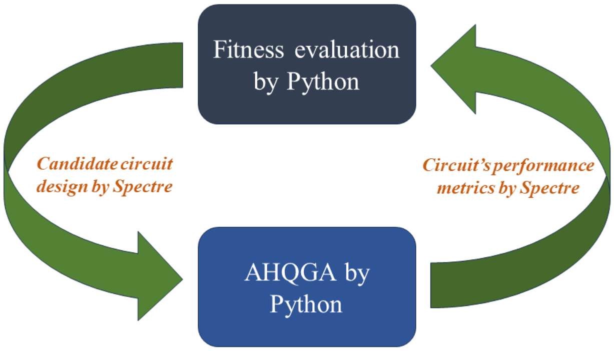


Figure 3: **Quantum rotation scheme to achieve the best individual.**



Fitness evaluation by Python
Candidate circuit design by Spectre
Circuit's performance metrics by Spectre
AHQGA by Python

Figure 4: **AHQGA-Spectre framework[2] with mutual interaction between Python and Spectre platforms.**

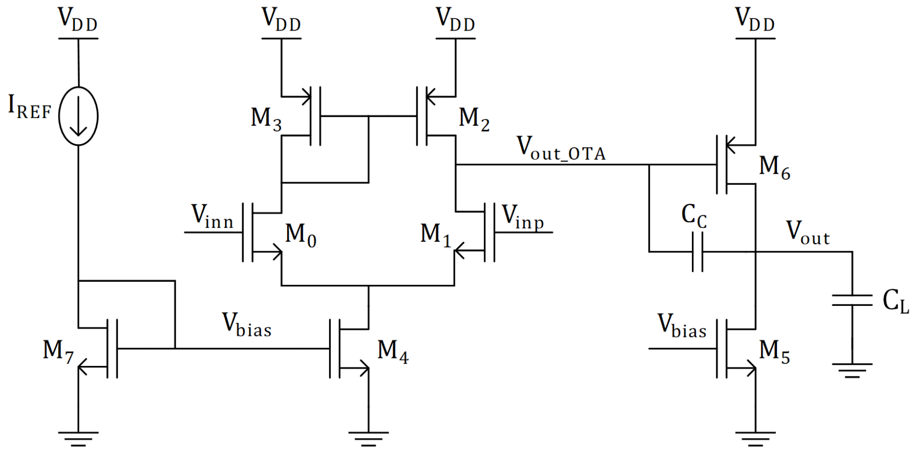


Figure 5: **Circuit schematic of the two-stage Miller-compensated op-amp.**

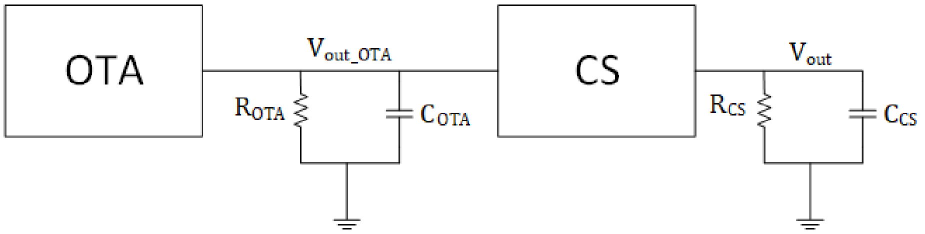


Figure 6: **Block diagram presentation for obtaining the two poles of the two-stage op-amp by inspection.**

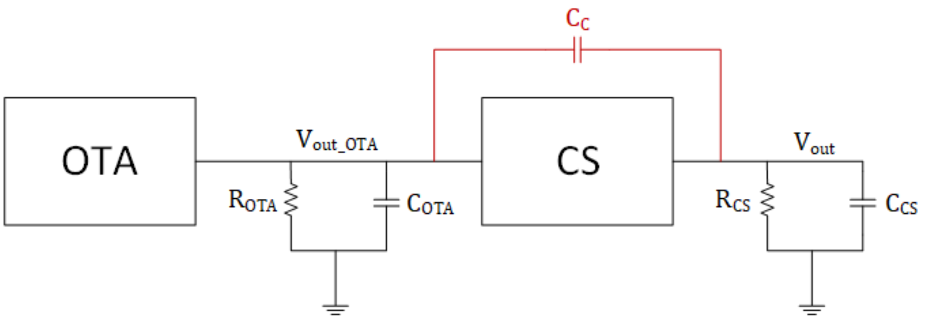


Figure 7: **Block diagram presentation for obtaining the two poles of the two-stage op-amp by inspection, with the additional compensation capacitor C_C.**

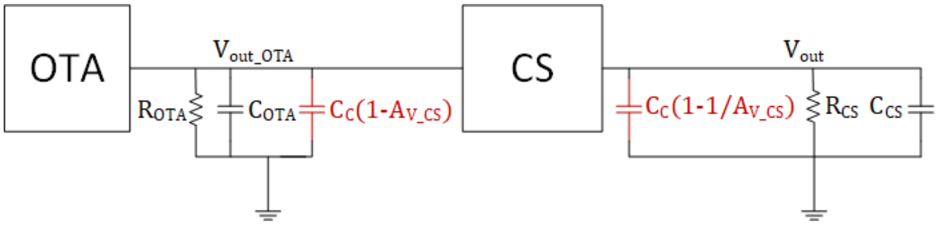


Figure 8: **Block diagram presentation for obtaining the two poles of the two-stage op-amp by inspection, with the Miller theorem applied for the compensation capacitor C_C.**

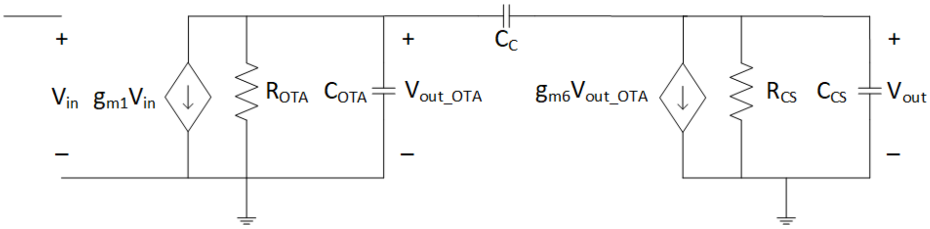


Figure 9: **Small-signal model of the two-stage Miller-compensated op-amp.**

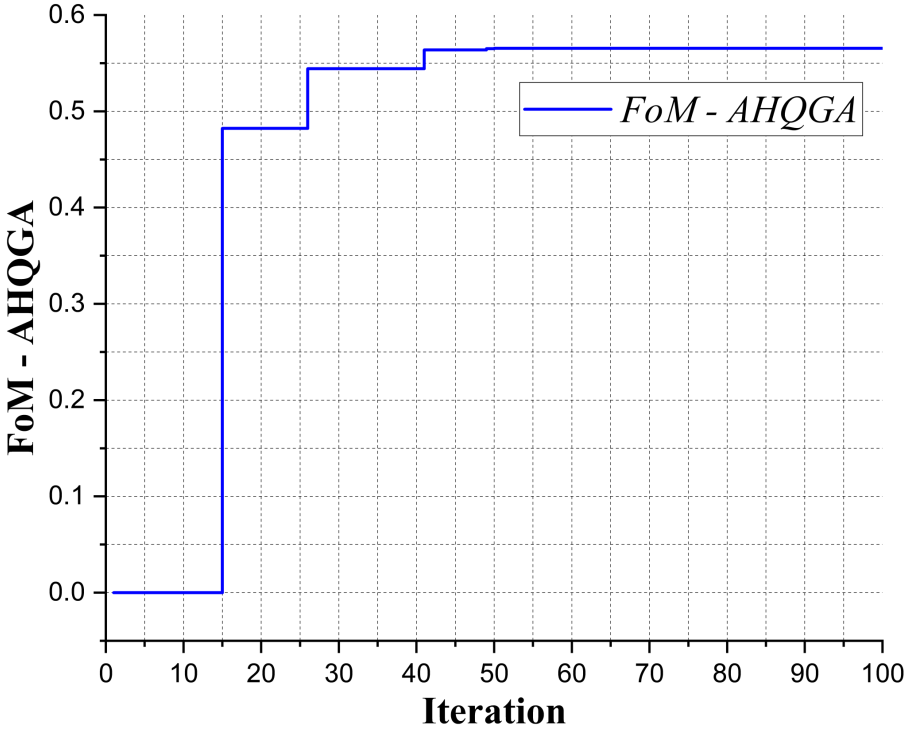


Figure 10: **FoM values by AHQGA versus 100 iterations.**

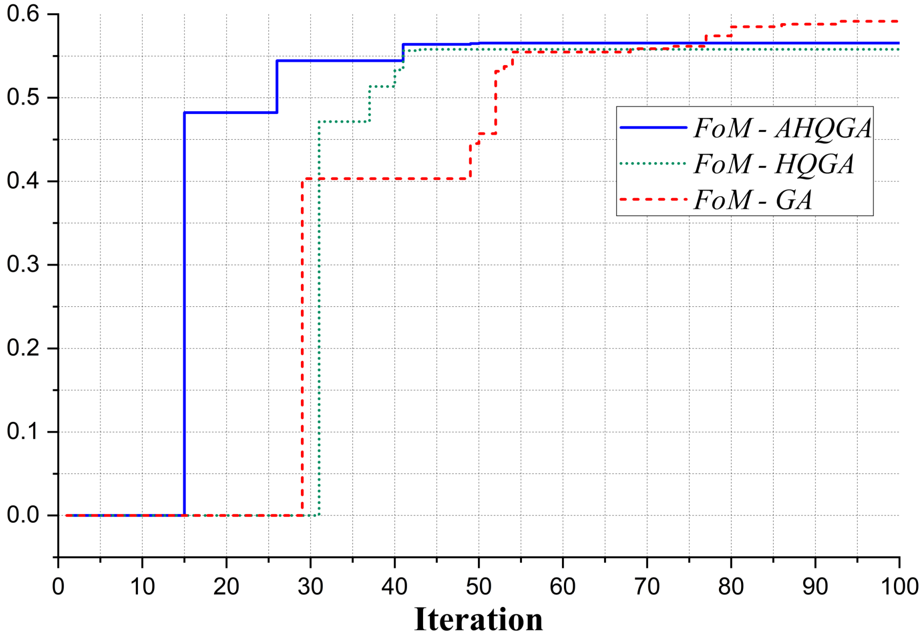


Figure 11: **FoM comparison between AHQGA, HQGA with fixed $\triangle\theta$, and GA over 100 iterations.**