

Particle Swarm Optimization of a Rail-to-Rail Delay Element for Maximum Linearity

Jordan Lee Gauci¹, Edward Gatt¹, Owen Casha¹, Giacinto De Cataldo², Ivan Grech¹, and Joseph Micallef¹

Abstract—This paper illustrates the use of the Particle Swarm Optimization (PSO) algorithm to maximize the linearity of a rail-to-rail delay element. Previous approaches relied on approximating the piecewise time-delay model of the delay element through either the Newton Polynomial or the Lagrange Polynomial methods. While adequate linearity was achieved in both cases, this could be further improved. This work successfully employed the PSO algorithm to improve the linearity by reducing the mean square error such that the delay element exhibits a spurious-free dynamic range of 29.62 dB, with a delay range of 170.4 ns. The results were verified in Cadence using the X-FAB 0.18 μm technology.

I. INTRODUCTION

Even though digital circuits are nowadays widely used in many applications, analog circuits are still important, particularly in analog to digital converters, radio frequency circuits and amplifiers. Tools for automating and optimizing such designs are still under-utilized. Of particular importance is the sizing of components to meet certain requirements, such as power consumption and bandwidth. Automatic sizing in analog circuits is therefore important as it allows a circuit designer to rapidly design high performance circuits which meet the project’s requirements [1].

This work deals with an implementation of the particle swarm optimization (PSO) technique that has been applied to optimize a delay element to be used in the forthcoming upgrade of the High Momentum Particle Identification Detector (HMPID) at the CERN Large Hadron Collider (LHC).

HMPID is a triggered detector, where data transfer is initiated upon the reception of a trigger signal driving the sample-and-hold circuitry. The charge on the pads should be read at its peak such that an optimal signal-to-noise ratio is obtained [2].

In Runs 1 and 2 (2009 - 2018), the trigger signal used to arrive approximately 1.2 μs after a collision has occurred. After the second long shutdown period (2019-2021), HMPID will be making use of the Level-Minus (LM) trigger signal that arrives earlier, after approximately 700 ns. A highly accurate delay generator is therefore required such that the timing of the trigger signal can be fine-tuned. In addition, it is important to have a wide delay range with a linear and monotonic transfer characteristic. This implies that rail-to-rail operation is essential such that the delay range is the largest possible [3].

In [3] and [4] it was shown that a quasi-linear delay can be achieved by using approximation techniques to simplify the

highly complex polynomial model of the circuit. However, in [3] linearity was only achieved for a limited range of $V_{tn} \leq V_c \leq V_{DD} - V_{tn}$, where V_c is the control voltage. In this work a new approach is adopted in order to achieve maximum linearity through the use of machine optimization algorithms, specifically the PSO technique.

A brief overview of optimization algorithms is given in Section II together with the motivation for using the PSO algorithm. In Section III, the delay element circuit is introduced in some detail, while in Section IV the implementation of the PSO is given. Section V highlights the improvements achieved compared to previous works, and finally conclusions are drawn in Section VI.

II. OPTIMIZATION ALGORITHMS

There are two methods in which optimization can take place; either simulation-based, or equation-based. The former performs optimization on circuit parameters based on the simulation results. It takes care of any parasitic devices, if modeled, and can be used for any type of circuit with minimum setup. Unfortunately this method is more computationally expensive, as with each iteration the simulation needs to be re-evaluated. The second method is the equation-based approach. While the equations need to be derived for each type of circuit topology, this is done only once and the equation can be optimized using numerical solvers.

Optimization algorithms in integrated circuit design may be split into three main categories: Bio-inspired optimizations which encompass evolutionary algorithms and swarm intelligence algorithms, deterministic algorithms, and other optimization techniques that do not fit in the previous two categories such as simulated annealing, convex optimization, and greedy algorithms [5].

There are two most commonly used techniques for optimization of analog circuits through bio-inspired techniques; PSO, and Genetic Algorithm (GA). The PSO algorithm involves the use of an initial swarm (a random set of generated particles) which move in the design search space towards the required optimal solution. The information is shared between each member of the swarm. The GA, on the other hand, is built upon the principle of “survival of the fittest” and solutions are generated based upon experience and environment [6].

The PSO and GA are similar in the sense that they are both population-based search approaches and utilize information sharing between particles. In [6] it was shown that the PSO is more computationally efficient when compared to the GA, even though the former yields similar results to the GA

¹Department of Microelectronics and Nanoelectronics, University of Malta, Malta jordan-lee.gauci.10@um.edu.mt

²Istituto Nazionale di Fisica Nucleare, Sezione di Bari, Italy

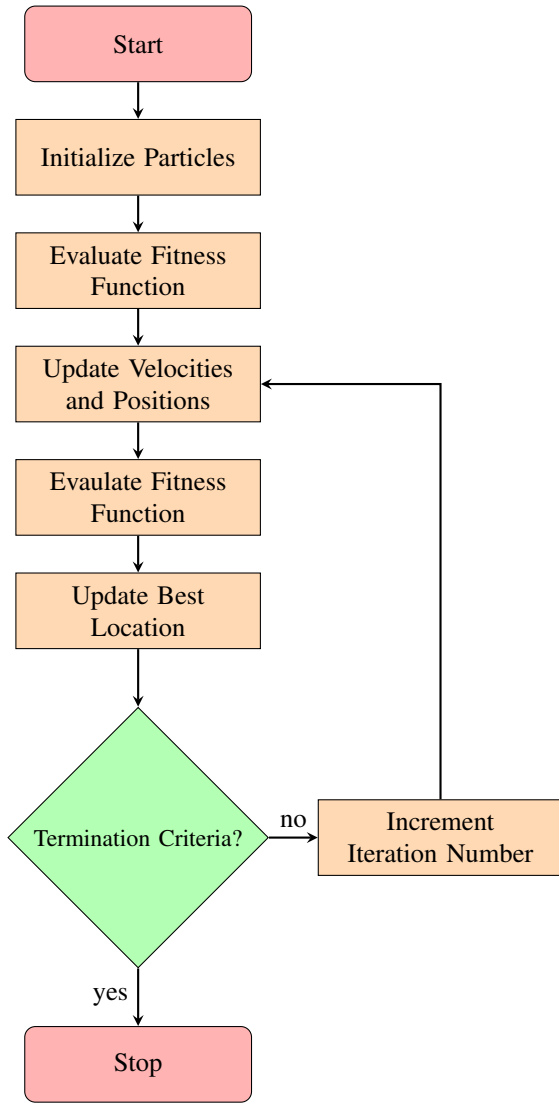


Fig. 1. Flowchart of the PSO algorithm.

approach. Thus, the PSO is a good algorithm to be able to optimize the aspect ratios of the transistors in a circuit.

A flowchart of the PSO algorithm is depicted in Fig. 1. Initially, the basic swarm is created, consisting of N particles, in the multi-dimensional search space, with D dimensions, where each dimension represents a variable in the circuit model. Each particle is characterized by a position and velocity. Once the particles have been created, and the iteration number set to 1, the objective function is calculated for each particle, and the fitness value of each particle is compared with the best particle value. On each iteration, the position and velocity of each particle are updated according to its own best position, P_{Best} , and the best position in the entire swarm G_{Best} . The iteration number is incremented and the objective function is calculated once again. This continues until the termination criteria (e.g. maximum number of iterations) is met [7].

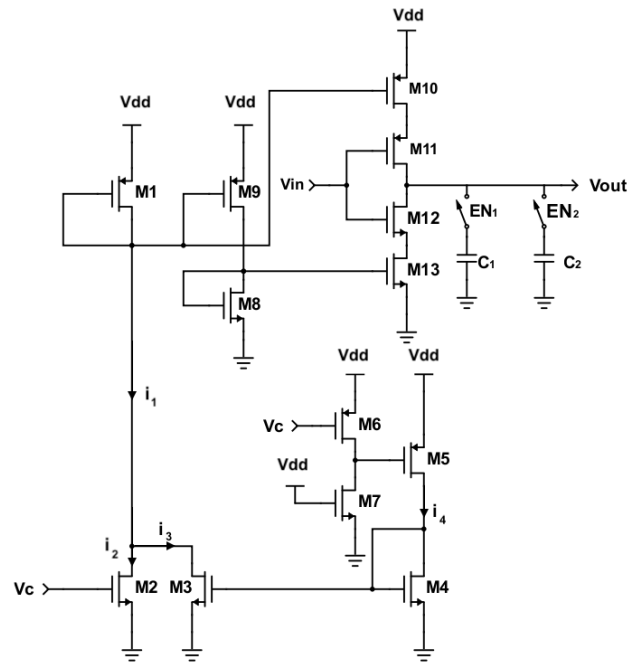


Fig. 2. Improved linear delay element circuit with extended programmable range and symmetric operation [4].

III. CASE STUDY: RAIL-TO-RAIL LINEAR DELAY ELEMENT

The delay element, based on the work proposed in [8] and improved in [4], is illustrated in Fig. 2. The circuit is based on a current-starved inverter architecture and can achieve both a quasi-linear delay and rail-to-rail operation. This is achieved through the addition of transistors $M_3 - M_7$. An inverting common-source amplifier (consisting of M_6 and M_7) is required to be able to achieve a monotonic and quasi-linear relationship in the delay response of the circuit. This would not have been possible if the tuning voltage, V_c , is applied directly on M_5 . The improvements performed in [4] allow for both edges of the input signal to be delayed, while also allowing for an extension of the range due to a programmable load capacitance. The control voltage V_c finely tunes the delay, while switches EN_1 and EN_2 increase the effective load capacitance at the output node and hence provide coarse control. In addition, this circuit also enables an increase in the delay range via proper scaling of the current mirror ratios M_{10}/M_1 and M_{13}/M_8 . This is particularly useful for limiting the size of the on-chip capacitors, C_1 and C_2 .

The delay time, T_d , in a current-starved inverter architecture is related to the current through Eq. 1, where I_c is the current in the delay element, C_L is the total load capacitance, and V_{DD} is the supply voltage [8, 9].

$$T_d \propto \frac{C_L}{I_c} V_{DD} \quad (1)$$

To model the rail-to-rail current in M_1 a piecewise expression needs to be considered to allow for the cases when

$V_c < V_{tn}$ and $V_c > V_{tn}$, where V_c is the control voltage, and V_{tn} is the threshold voltage of the NMOS transistor. Thus, assuming that M_2 and M_3 remain in pinch-off, the piecewise expression for the current in M_1 is given by:

$$i_1 = \begin{cases} i_3 & V_c < V_{tn} \\ i_2 + i_3 & V_c \geq V_{tn} \end{cases} \quad (2)$$

where i_x is the current through transistor M_x and x is the transistor identifier.

When M_2 and M_3 operate in pinch-off region, the currents are given by:

$$i_2 = \frac{K'_n W_2}{2 L_2} (V_c - V_{tn})^2 \quad (3)$$

$$i_3 = \frac{\frac{W_3}{L_3}}{\frac{W_4}{L_4}} i_4 \quad (4)$$

and $i_4 = i_5$ is given by

$$i_4 = \frac{K'_p W_5}{2 L_5} (V_{sg5} - V_{tp})^2 \quad (5)$$

where

$$V_{sg5} = V_N - \sqrt{(V_N)^2 - 2 \frac{K_p}{K_n} (V_{DD} - V_c - V_{tp})^2} \quad (6)$$

and $V_N = V_{DD} - V_{tn}$, $K_p = K'_p \frac{W_6}{L_6}$ and $K_n = K'_n \frac{W_7}{L_7}$.

This means that by Eq. 1, the delay that can be generated may be modelled as a piecewise equation in the form of Eq. 7.

$$T_d \propto \begin{cases} \frac{C_L}{i_3} V_{DD} & V_c < V_{tn} \\ \frac{C_L}{i_2 + i_3} V_{DD} & V_c \geq V_{tn} \end{cases} \quad (7)$$

IV. IMPROVING THE LINEARITY OF THE RAIL-TO-RAIL DELAY ELEMENT

There are two methods that can be used to improve the linearity of the delay element. The first method is to approximate the piecewise equation of Eq. 1 by using either the Lagrange Polynomial or Newton Polynomial methods, as described in [3] and [4], respectively. While these methods can achieve good linearity, the process is cumbersome as this is done manually, while introducing a certain error due to the nature of the approximation techniques used. The second method involves the use of bio-inspired numerical optimization techniques to find automatically the values of the aspect ratios of the transistors which yield the highest linearity.

A. Objective Function

To find a measure of the linearity of the time delay equation, the mean square error (MSE) can be considered. The MSE, σ is given by:

$$\sigma = \frac{1}{n} \sum_{i=1}^n (T_{di} - \hat{T}_{di})^2 \quad (8)$$

where n is the number of samples considered, T_d is the time delay of the circuit according to the model, and \hat{T}_d is the perfect linear time delay, given by Eq. 9, where R is the required resolution, and O is the offset. The gradient is negative since the delay is inversely proportional to the current, and hence inversely proportional to V_c .

$$\hat{T}_d = -RV_c + O \quad (9)$$

The MSE is the objective function of the optimization algorithm. In other words, to obtain the most linear delay, the MSE needs to be minimized. For the HMPID application considered in this work, the required delay element should have a resolution of -90 ns/V with an offset (delay value at $V_c = 0$ V) of around 360 ns.

B. Variables and their Constraints

In this case-study the variables are the aspect ratios of transistors $M_2 - M_7$. To ensure that the values of the sizes of the transistors can be realized, the constraints were chosen specifically to keep the area to a minimum. As such, the lower bound values of the aspect ratio was set to 0.1, while the upper bound value was set to 30.

C. Implementation of the PSO Algorithm

The PSO algorithm was implemented in MATLAB. Three functions were created for proper code re-usability and readability. The main function consists of the primary PSO algorithm, while the other two functions evaluate the delay model, and the ideal linear delay model. The former takes as its parameters the aspect ratios of the transistors and the control voltage vector, and it works out the time-delay model according to Eq. 7. The latter takes as its parameters the control voltage vector, and the required gradient and offset, and evaluates Eq. 9.

In the main function, the parameters of the PSO algorithm are defined. Specifically these parameters are the swarm size, N , and the number of dimensions (variables), D . In this scenario, there are six unknown parameters which are the aspect ratios of transistors $M_2 - M_7$. The maximum and minimum inertia weights, w_{max} and w_{min} respectively, are set in this part of the code. The inertia weights are multipliers for the current velocity of the particles such that the new positions may be updated.

To obtain the most reliable results, the main program runs for a number of times. As such, the main function consists of two nested loops, where the first loop takes care of the run, and the second loop is the principal iteration of the PSO algorithm.

The position of the particles is first initialized through Eq. 12, where UB(j) and LB(j) denote the upper bounds and lower bounds of each dimension, j , and r is a random number generated between 0 and 1. The initial velocity of the particles is set by multiplying the initial position by a factor of 0.1.

$$X_{i,j} = \text{Round}(\text{LB}(j) + r \times (\text{UB}(j) - \text{LB}(j))) \quad (12)$$

$$V_{i,j}^{k+1} = w \times V_{i,j}^k + c_1 \times r_{1,i,j} \times (P_{Best_{i,j}}^k - X_{i,j}^k) + c_2 \times r_{2,i,j} \times (G_{Best_j}^k - X_{i,j}^k) \quad (10)$$

$$X_{i,j}^{k+1} = X_{i,j}^k + V_{i,j}^{k+1} \quad (11)$$

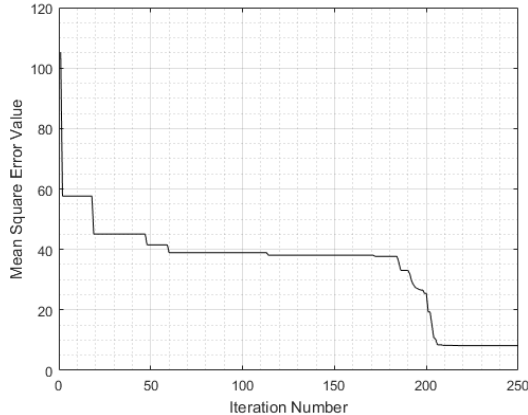


Fig. 3. Convergence profile of the PSO algorithm for the best run.

The script then performs a check to ensure that the value of each of the generated particles lies within the bounds.

The values of the particles are passed to the model function, such that the delay model for each particle can be calculated, and the MSE for each particle is evaluated. For each iteration, the minimum MSE is found and its value saved, together with the initial best particle and best global particle.

The first step in the inner-loop is to update the inertial weight, w , and then calculate the new velocities vector, V , of the particles according to Eq. 10, and the position vector, X , of each particle according to Eq. 11. In these equations, k denotes the iteration number, and i represents the particle number. Parameters $r_{1,i,j}, r_{2,i,j}$ refer to a random number generated in the range 0 to 1, while c_1, c_2 are acceleration factors.

With every iteration, the values of the particles are re-checked through the model function, and the MSE is recalculated. The least MSE value for each particle (denoted by P_{Best}) and for each swarm (G_{Best}) are stored separately. While the iteration number has not reached the maximum number of iterations, the inner loop restarts. Once the maximum number of iterations has been reached, the run counter of the outer loop is increased, and the algorithm is re-executed.

When the program is complete, the best MSE value obtained is presented together with the relevant aspect ratios of the transistors that yielded that result.

V. VERIFICATION OF MODEL AND COMPARISON

The PSO algorithm presented in the previous section was executed and the optimal aspect ratios of the transistors were found. The algorithm was executed with six variables, and a swarm population of 100. The minimum and maximum

TABLE I
ASPECT RATIO OF TRANSISTORS.

Transistor Identifier	Original Work [8]	Previous Work [4]	This Work
M_2	0.9	0.1	0.1
M_3	3.9	12	17
M_4	11.5	12	30
M_5	12.3	1	1.8
M_6	17.7	5.9	29.7
M_7	10.8	4.85	22.2

inertia weights were set to 0.4 and 1.4, respectively. Acceleration factors c_1 and c_2 were set to 1.5 and 2, with an initial velocity of 0.1.

The time it takes the program to find the optimal solution depends on the swarm size, number of iterations, and number of runs. In this case, the algorithm was executed with 250 iterations and 100 independent runs, and the optimal solution (being the solution that yields the minimum MSE) took approximately 520 seconds, while running on a Core i7-4790 processor at 3.6 GHz, with 10 GB of memory. Fig. 3 illustrates the convergence profile of the PSO algorithm for the best run.

The results are presented in Table I, together with the transistor values used in the original work [8] and those presented in the previous work [4]. For the values obtained, the MSE is 8.16, while for the previous work this value was 12.8. This implies that an improvement in the linearity of the delay element was achieved.

The circuit illustrated in Fig. 2 was implemented in Cadence, which is an electronic design automation tool, using the 0.18 μm X-FAB technology, with the optimized transistor aspect ratios. The delay versus control voltage characteristic obtained from Cadence is presented in Fig. 4, together with the ideal delay response. It can be seen that there is a good correspondence between the analytical model and the simulation results obtained from Cadence, using a 1 pF load.

To verify the linearity of the delay element, a sinusoidal input was applied to the V_c terminal of the circuit, and the delay between the input and output square waves was calculated through another MATLAB script. The sinusoidal input has a frequency of 2.148 kHz, while the input square wave has a value of 200 kHz. The sampling frequency used was 2 GHz, with a simulation time of 5.125 ms. The sinusoidal control voltage input and the delay response at the output are plotted in Fig. 5.

The frequency spectrum of the generated delay is plotted in Fig. 6. The spurious-free dynamic range (SFDR) of the delay is equal to 29.62 dB. This implies that there is an improvement of 4.5 dB over the work presented in [4]. While this value is still less than that achieved by the authors in [8], where the SFDR was equal to 35 dB, the range of this

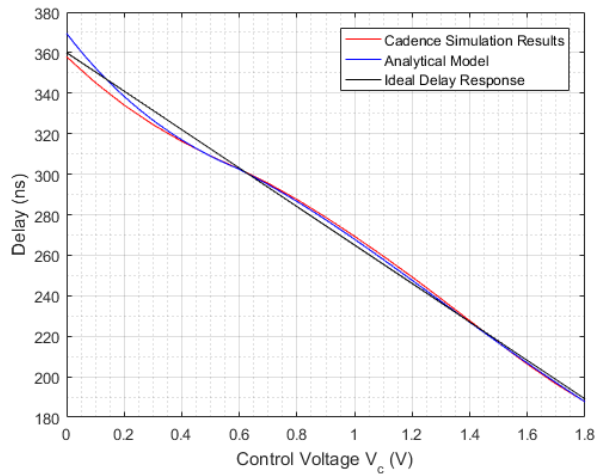


Fig. 4. Comparison of the simulation results obtained from Cadence with the results generated from the analytical model and the ideal delay response.

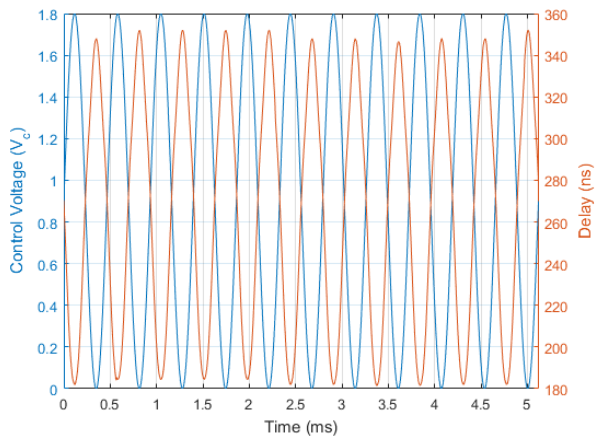


Fig. 5. Transient response of the delay element for a sinusoidal control voltage applied at the input.

delay element is much wider (170.4 ns), compared to 1.4 ns achieved in [8]. The simulated signal-to-noise ratio (SNDR) is 25.6 dB, an improvement of 2.56 dB over the work in [4].

VI. CONCLUSION

This paper has presented the application of the PSO algorithm to improve the linearity of a rail-to-rail delay element. Linearity was optimized by minimizing the MSE function between the analytical model of the delay and the ideal case. The PSO algorithm finds the optimal value of the transistor aspect ratios such that maximum linearity could be achieved. The use of the mathematical model was specifically chosen, as optimization of circuit parameters based on simulation is much more computationally expensive. The optimized design was validated via Cadence simulations using the X-FAB 0.18 μm technology. These simulations show that this method yields an improvement of the SFDR and SNDR, over those obtained in previous works, by 4.5 dB and 2.56 dB respectively.

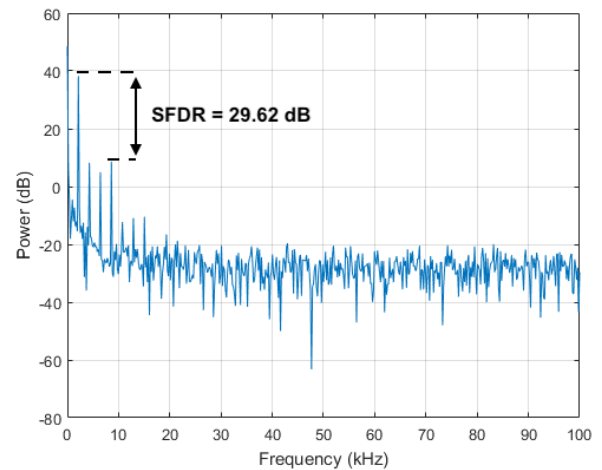


Fig. 6. Frequency spectrum of the generated delay for $C_L = 1 \text{ pF}$.

ACKNOWLEDGEMENTS

The research work disclosed in this publication is funded by the ENDEAVOUR Scholarship Scheme (Malta). The scholarship may be part-financed by the European Union - European Social Fund (ESF) under Operational Programme II - Cohesion Policy 2014-2020, "Investing in human capital to create more opportunities and promote the well being of society."

REFERENCES

- [1] M. Fakhfakh, Y. Cooren, A. Sallem, M. Loulou, and P. Siarry, "Analog circuit design optimization through the particle swarm optimization technique," *Analog Integrated Circuits and Signal Processing*, vol. 63, no. 1, pp. 71–82, 2010.
- [2] J. L. Gauci, E. Gatt, G. De Cataldo, O. Casha, and I. Grech, "An Analytical Model of the Delay Generator for the Triggering of Particle Detectors at CERN LHC," in *2017 IEEE New Generation of CAS (NGCAS)*, 2017, pp. 69–72.
- [3] J. L. Gauci, E. Gatt, O. Casha, G. De Cataldo, and I. Grech, "On the Design of a Linear Delay Element for the Triggering Module at CERN LHC," in *14th IEEE Conference on PhD Research in Microelectronics and Electronics (PRIME)*, 2018.
- [4] —, "Design of a Quasi-Linear Rail-to-Rail Delay Element with an Extended Programmable Range," in *Electronics, Circuits and Systems (ICECS), 2018 25th IEEE International Conference on*, 2018.
- [5] S. V. Kumar, P. Rao, H. Sharath, B. Sachin, U. Ravi, and B. Monica, "Review on vlsi design using optimization and self-adaptive particle swarm optimization," *Journal of King Saud University-Computer and Information Sciences*, 2018.
- [6] R. Hassan, B. Cohanım, O. De Weck, and G. Venter, "A comparison of particle swarm optimization and the genetic algorithm," in *46th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference*, 2005, p. 1897.
- [7] P. Kumar and K. Duraiswamy, "An optimized device sizing of analog circuits using particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks, Nov. 27-Dec. 1, IEEE Xplore*. Citeseer, 2012.
- [8] H. Rivandi, S. Ebrahimi, and M. Saberi, "A low-power rail-to-rail input-range linear delay element circuit," *AEU-International Journal of Electronics and Communications*, vol. 79, pp. 26–32, 2017.
- [9] E. Zafarkhah, M. Maymandi-Nejad, and M. Zare, "Improved accuracy equation for propagation delay of a CMOS inverter in a single ended ring oscillator," *AEU-International Journal of Electronics and Communications*, vol. 71, pp. 110–117, 2017.