Proceedings of Joint Conference of
11th International Science, Social Sciences, Engineering and Energy Conference (I-SEEC 2022) and,
6th International Conference on Technology and Social Science 2022 (ICTSS 2022)

High-Resolution Unary DAC Unit Cell Sorting Algorithms for Linearity Improvement with Measured Unit Cell Values

Kaiki Okabe [1,a], Shogo Katayama [1,b], Anna Kuwana[1,c]
Haruo Kobayashi[1,d]

[1] Division of Electronics and Informatics, Faculty of Science and Technology, Gunma University
1-5-1 Tenjin-cho, Kiryu, Gunma, 376-851, Japan

[a]< t190d027@gunma-u.ac.jp >, [b]< t202d601@gunma-u.ac.jp >, [c]< kuwana.anna@gunma-u.ac.jp >
[d]<koba@gunma-u.ac.jp>

Keywords: Unary DAC, Mismatch, Linearity, Sorting

Abstract. This paper describes the linearity self-calibration method for the unary digital-to-analog converter (DAC), by measuring the unit current values and arranging their selection order so that the integral nonlinearity is small. The total number of their possible order selection is very large for a high-resolution unary DAC and their integral nonlinearity (INL) calculation may not be done within reasonable time if all of combinations are calculated. Hence, we have investigated the reduction of number with several methods, and proposed an effective algorithm based on the consideration how the selection order can realize the DAC INL reduction.

1. Introduction

The high-speed high-resolution DAC is used for the transmitter and the high-speed high-frequency electronic measurement instrument. There high linearity is required for good spurious free dynamic range (SFDR) [1, 2]. However, in advanced nano-CMOS process, small semiconductor devices on silicon wafers suffer from random and systematic mismatches regarding the characteristics of MOSFETs, resistors, and capacitors [3], which make the DAC input and output relationship non-linear. There analog performance may be deteriorated, but digital circuit is almost free; so digitally assisted analog technology is attractive to improve the analog circuit performance.

We investigate here the linearity self-calibration method for the unary DAC, with measured unit current values and by arranging their selection order so that the INL is small. The total number of their possible order selection is very large and their INL calculation may not be done within reasonable time. Then we investigated the reduction of number with several methods using numerical simulations and proposed an effective algorithm based on the consideration how the selection order can realize the DAC INL reduction.

Notice that we have investigated similar algorithms in [4,5], with the assumption that only the order of unit current values is known using current value comparison in a digital method. However, here we assume that all the unit current values are known by measurement using a slow-yet-accurate ADC (such as a delta-sigma ADC). There are many other methods to improve the unary DAC linearity under different problem formulations and the reader can refer to [6-15].

2. Problem Formulation

Let us consider a 4-bit unary DAC in Fig. 1, as an example. It consists of 15 unit current sources with current switches, a resistor (R) and a binary-to-thermometer decoder. Ideally, the values of all the current sources are identical.

$$I_1 = I_2 = I_3 = I_4 = I_5 = I_6 = I_7 = I_8 = I_9 = I_A = I_B = I_C = I_D = I_E = I_F$$

When the digital input (Din) is N, N current switchers are ON while the others are OFF, and the current of N I flows through the resistor R and the output analog voltage (Vout) is N RI. Here, N=0, 1, 2, 3, …, 15. In other words, the output voltage is proportional to the digital input. However, in reality, the values of the currents can be mismatched due to process variation, and the followings are assumed:

$$I_1 = I + \Delta I_1 \;\; I_2 = I + \Delta I_2 \;\; I_3 = I + \Delta I_3 \;\; I_4 = I + \Delta I_4$$
$$I_5 = I + \Delta I_5 \;\; I_6 = I + \Delta I_6 \;\; I_7 = I + \Delta I_7 \;\; I_8 = I + \Delta I_8$$
$$I_9 = I + \Delta I_9 \;\; I_A = I + \Delta I_A \;\; I_B = I + \Delta I_B \;\; I_C = I + \Delta I_C$$
$$I_D = I + \Delta I_D \;\; I_E = I + \Delta I_E \;\; I_F = I + \Delta I_F$$

I is defined as the average current value:

$$I := \frac{1}{15}\,(I_1 + I_2 + I_3 + I_4 + I_5 + I_6 + I_7 + I_8 + I_9 + I_A + I_B + I_C + I_D + I_E + I_F)$$

Notice that the following is derived from the above equations:

$$\Delta I_1 + \Delta I_2 + \Delta I_3 + \Delta I_4 + \Delta I_5 + \Delta I_6 + \Delta I_7 + \Delta I_8 + \Delta I_9 + \Delta I_A + \Delta I_B + \Delta I_C + \Delta I_D + \Delta I_E + \Delta I_F = 0$$

Then we have the relationships among digital input, analog output and integral nonlinearity (INL) of the 4-bit current-steering unary DAC, as shown in Fig. 2.

We observe that the unary DAC has some redundancy. For example, as shown in Fig. 3, there are many operation possibilities of the unary DAC in case that the digital input is 4. We see that their INL can be different depending on the mismatches $\Delta I_1, \Delta I_2, \ldots, \Delta I_F$ and the unit cell selection.

- In Fig. 3 (a), S_1, S_2, S_3, S_4 are ON and INL(4) is $\Delta I_1 + \Delta I_2 + \Delta I_3 + \Delta I_4$.
- In Fig. 3 (b), S_5, S_6, S_7, S_8 are ON and INL(4) is $\Delta I_5 + \Delta I_6 + \Delta I_7 + \Delta I_8$.
- In Fig. 3 (c), S_B, S_C, S_D, S_E are ON and INL(4) is $\Delta I_B + \Delta I_C + \Delta I_D + \Delta I_E$.

In other words, INL can be reduced by selecting appropriate unit current cells.

Now let us consider to arrange the order of $I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_8, I_9, I_A, I_B, I_C, I_D, I_E, I_F$ to $I_a, I_b, I_c, I_d, I_e, I_f, I_g, I_h, I_i, I_j, I_k, I_l, I_m, I_n, I_p$.
In case that the digital input is 4, S_a, S_b, S_c, S_d are ON and INL is $\Delta I_a + \Delta I_b + \Delta I_c + \Delta I_d$.

For example, suppose that (a b c d e f g h i j k l m n p) = (5 A 3 9 8 B C 1 D 2 E 6 F 4 7).
Then in case that the digital input is 4, S_5, S_A, S_3, S_9 are ON and INL(4) is $\Delta I_5 + \Delta I_A + \Delta I_3 + \Delta I_9$.

Now our problem is formulated as follows:
Suppose that all values of $I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_8, I_9, I_A, I_B, I_C, I_D, I_E, I_F$ are known (by measuring them such as a delta-sigma ADC). Then we choose the unit cell selection order $I_a, I_b, I_c, I_d, I_e, I_f, I_g, I_h, I_i, I_j, I_k, I_l, I_m, I_n, I_p$ so that the evaluation function J is minimum.

$$J = INL(0)^2 + INL(1)^2 + INL(2)^2 + INL(3)^2 + INL(4)^2 + INL(5)^2 + INL(6)^2 + INL(7)^2 + INL(8)^2 + INL(9)^2 + INL(A)^2 + INL(B)^2 + INL(C)^2 + INL(D)^2 + INL(E)^2 + INL(F)^2 \rightarrow \text{minimum}$$

The number of possible selection orders is 15! in 4-bit case. However, say, in 8-bit case, it is 255! which is too large to compute each corresponding J within reasonable time in-field. Then we consider to obtain sub-optimal J (close to minimum, not necessarily minimum) with reduced number of possible orders; this attempt is described in the next section.

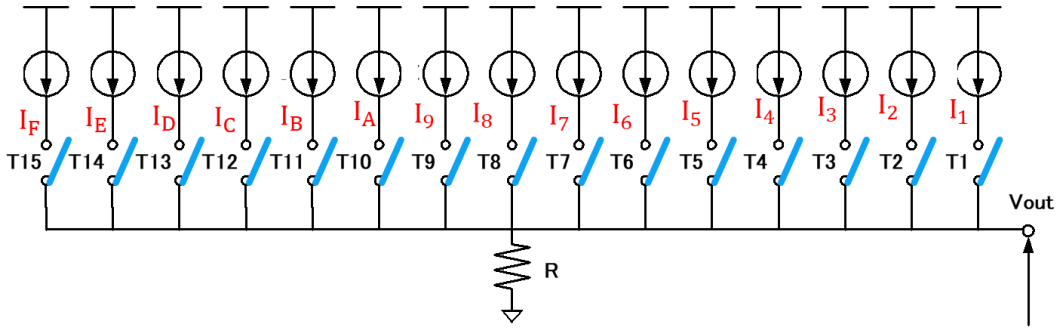

Fig. 1. 4-bit Current-Steering Unary DAC (binary-to-thermometer decoder is not shown).

Digital Input	Analog Output	Integral Nonlinearity (INL)
0	Vout(0)=0	INL(0)=0
1	Vout(1)=R I1	INL(1)=R[ΔI1]
2	Vout(2)=R[I1+I2]	INL(2)=R[ΔI1+ΔI2]
3	Vout(3)=R[I1+I2+I3]	INL(3)=R[ΔI1+ΔI2+ΔI3]
4	Vout(4)=R[1+I2+I3+I4]	INL(4)=R[ΔI1+ΔI2+ΔI3+ΔI4]
5	Vout(5)=R[I1+I2+I3+I4+I5]	INL(5)=R[ΔI1+ΔI2+ΔI3+ΔI4+ΔI5]
6	Vout(6)=R[I1+I2+I3+I4+I5+I6]	INL(6)=R[ΔI1+ΔI2+ΔI3+ΔI4+ΔI5+ΔI6]
7	Vout(7)=R[I1+I2+I3+I4+I5+I6+I7]	INL(7)=R[ΔI1+ΔI2+ΔI3+ΔI4+ΔI5+ΔI6+ΔI7]
8	Vout(8)=R[I1+I2+I3+I4+I5+I6+I7+I8]	INL(8)=R[ΔI1+ΔI2+ΔI3+ΔI4+ΔI5+ΔI6+ΔI7+ΔI8]
9	Vout(9)=R[I1+I2+I3+I4+I5+I6+I7+I8+I9]	INL(9)=R[ΔI1+ΔI2+ΔI3+ΔI4+ΔI5+ΔI6+ΔI7+ΔI8+ΔI9]
10	Vout(10)=R[I1+I2+I3+I4+I5+I6+I7+I8+I9+IA]	INL(10)=R[ΔI1+ΔI2+ΔI3+ΔI4+ΔI5+ΔI6+ΔI7+ΔI8+ΔI9+ΔIA]
11	Vout(11)=R[I1+I2+I3+I4+I5+I6+I7+I8+I9+IA+IB]	INL(11)=R[ΔI1+ΔI2+ΔI3+ΔI4+ΔI5+ΔI6+ΔI7+ΔI8+ΔI9+ΔIA+ΔIB]
12	Vout(12)=R[I1+I2+I3+I4+I5+I6+I7+I8+I9+IA+IB+IC]	INL(12)=R[ΔI1+ΔI2+ΔI3+ΔI4+ΔI5+ΔI6+ΔI7+ΔI8+ΔI9+ΔIA+ΔIB+ΔIC]
13	Vout(13)=R[I1+I2+I3+I4+I5+I6+I7+I8+I9+IA+IB+IC+ID]	INL(13)=R[ΔI1+ΔI2+ΔI3+ΔI4+ΔI5+ΔI6+ΔI7+ΔI8+ΔI9+ΔIA+ΔIB+ΔIC+ΔD]
14	Vout(14)=R[I1+I2+I3+I4+I5+I6+I7+I8+I9+IA+IB+IC+ID+IE]	INL(14)=R[ΔI1+ΔI2+ΔI3+ΔI4+ΔI5+ΔI6+ΔI7+ΔI8+ΔI9+ΔIA+ΔIB+ΔIC+ΔID+ΔIE]
15	Vout(15)=15 R I	INL(15)=0

Fig. 2. Digital input, analog output and INL of a 4-bit current-steering unary DAC.

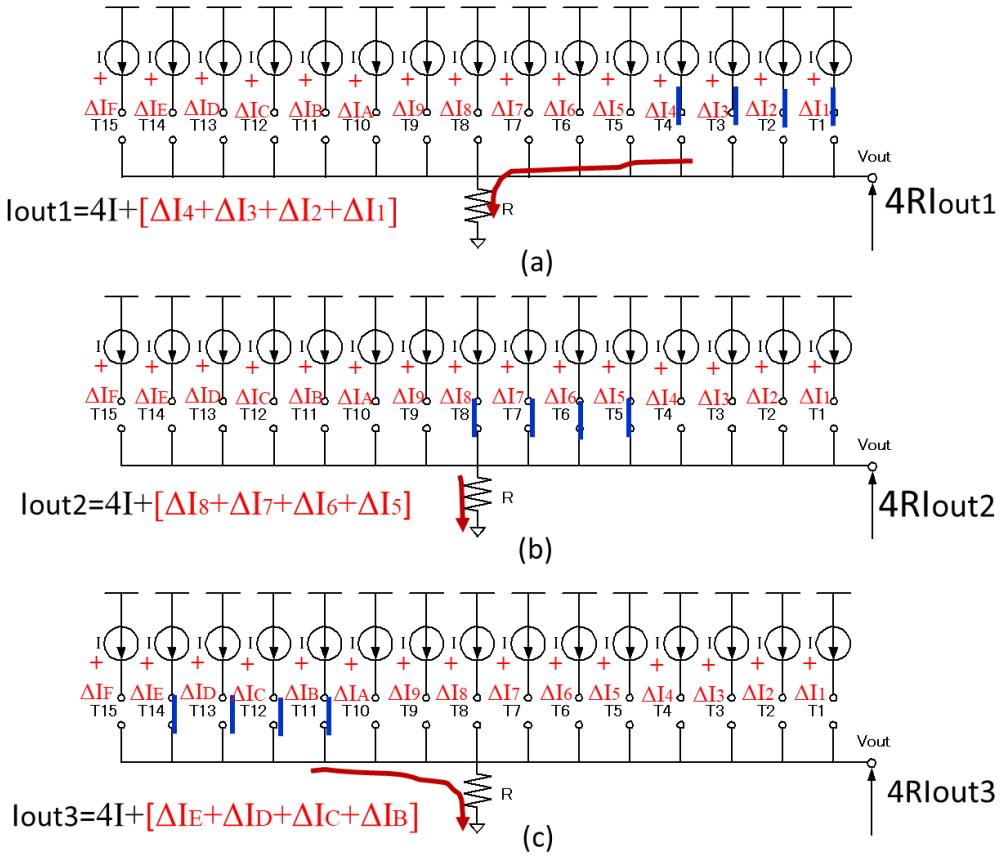

Fig. 3. Operation possibilities of a 4-bit current-steering unary DAC in case that digital input is 4.

3. Unit Cell Selection Algorithm

Now we investigate the unit cell selection algorithm for the unary DAC with reduced number of possible orders in order to make its INL small.

Let us consider the case of 4 current sources (I_1, I_2, I_3, I_4) with variations.

$$I_1 = I + \Delta I_1,\ \ I_2 = I + \Delta I_2,\ \ I_3 = I + \Delta I_3,\ \ I_4 = I + \Delta I_4.$$

We define their average as follows:

$$I\ \ := \frac{1}{4}(I_1 + I_2 + I_3 + I_4)$$

Table 2 shows different selection order cases and we see that their INLs can be different.
We found in simulation that INL evaluation function J becomes small when the selection order (a, b, c, d) is chosen such that the plus or minus sign of ΔI_a, ΔI_b, ΔI_c, ΔI_d is alternate and $|\Delta I_a|$, $|\Delta I_d|$ are small; this tendency was valid also in case of 12 current sources.

Table 2. INL in case of two unit-cell selection orders in a unary DAC with 4 current sources

Digital Input	Current Output	INL
1	I_1	ΔI_1
2	$I_1 + I_2$	$\Delta I_1 + \Delta I_2$
3	$I_1 + I_2 + I_3$	$\Delta I_1 + \Delta I_2 + \Delta I_3$
4	$I_1 + I_2 + I_3 + I_4$	$\Delta I_1 + \Delta I_2 + \Delta I_3 + \Delta I_4$

Digital Input	Current Output	INL
1	I_2	ΔI_2
2	$I_2 + I_1$	$\Delta I_2 + \Delta I_1$
3	$I_2 + I_1 + I_4$	$\Delta I_2 + \Delta I_1 + \Delta I_4$
4	$I_2 + I_1 + I_4 + I_3$	$\Delta I_2 + \Delta I_1 + \Delta I_4 + \Delta I_3$

3.1 Preliminary Investigation

We have tried 16 or more current sources cases.
1) Sequential change of the unit cell selection order at each step.
2) Random change of the unit cell selection order at each step.
3) Choose a pair of current sources and alternate them.

Simulation I:
- Initial state of the first trial: ΔI_k is arranged in an ascending order.
- Initial state of n-th trial (n>1): ΔI_k order arrangement is the same as the final state of (n-1)-th trial.
-Number of calculated combinations: 20x10^6
-Number of trials: 10
- rand() function in program C is used for random number generation
We choose the ΔI_k order arrangement where J is minimum in all 10 trials.

Our simulation results are as follows:
Regarding to the calculation speed, 1) is the fastest while 2) is the slowest.
Regarding to the minimum J, 2) is the smallest (0.008900), and 3) is the second best (0.009100) but comparable to 2), while 1) is the largest (0.213400).

Simulation II:
- srand() function in program C is used, which changing its "seed" value at each trial.
- Initial state of n-th trial (n>1): ΔI_k order arrangement is the same as the state of (n-1)-th trial where the smallest J is obtained.
-Number of calculated combinations: 20x10^6
- Number of trials: 21
- Number of "seed" values: 5
- Random change of the unit cell selection order at each step.

Program execution speed becomes slow but complete random number can be generated by changing its "seed" value, compared to rand() function.

Simulation result shows that J becomes 0.0320 from 0.2872.

Simulation III:
We set the following simulation termination conditions:
1) Maximum simulation time *T*max
2) Critical evaluation function value *J*crit
The simulation of the current source swap is terminated when the simulation time reaches *T*max or the evaluation function value becomes equal to or less than *J*crit.

- Number of simulation time: 15
- Maximum simulation time *T*max: 120s or 600s
- Critical evaluation function value *J*crit : one hundred-th of J in the initial condition.
- Initial condition: Ascendent order of current source values or
  One pair current source swap after ascendent order of current sources
- Current source swap method: Sequential change of the unit cell selection order at each step.

Case 1) *T*max:=120s, ascendant order of current source values

J=0.0588

Case 2) *T*max:=120s, one pair current source swap after ascendent order of current sources

J=0.0207

Case 3) *T*max:=600s, ascendant order of current source values

J=0.0283

Case 4) *T*max:=600s, one pair current source swap after ascendent order of current sources

J=0.0103

We see that the initial condition is important as well as the simulation time to obtain small J.

However, these approaches take a long calculation time and are not very effective.

3.2 Proposed Algorithm

We have observed that for INL reduction, the current source sorting, so as to cancel plus and minus current source mismatches as shown in Fig. 4 and in [16]. We consider that utilizing such knowledge leads to effective search for minimizing J. Our proposed algorithm is as follows:

1. First, select the current cell whose value is the closest to the average value I.
2. Randomly select the one that changes the sign of the accumulated current mismatches when its current mismatch is added.
3. Repeat 2 while there is current cell left that can change the sign of the accumulated current mismatches when its current mismatch is added.
4. When there is no current cell left which changes the sign during the calculation, the calculation is stopped and returns to 1.
5. Once all current cells are selected, calculate the evaluation function J.

Some cases calculated according to the above algorithm are shown below.

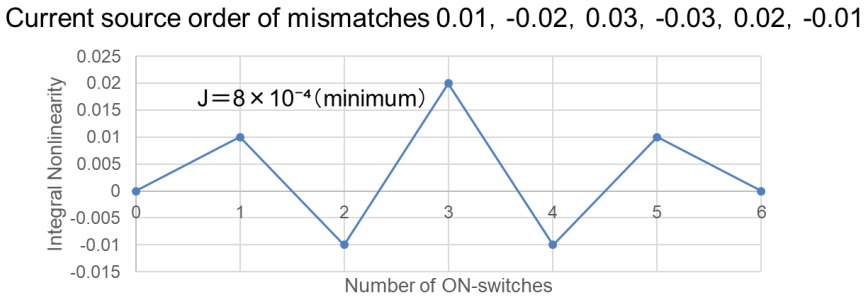

Fig. 4. Current source selection order to minimize INL, where plus and minus current source mismatches are assigned alternatively.

Case 1: Current mismatches ΔI/I -0.03, -0.02, -0.01, 0.00, 0.01, 0.02, 0.03

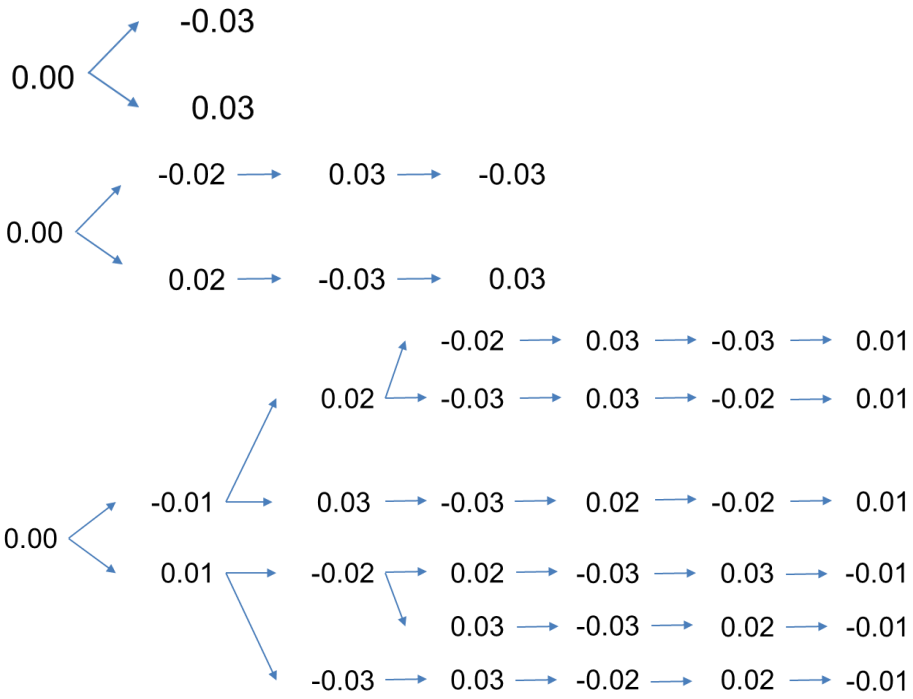

Fig. 5 Calculating process according to the algorithm in case 1.

There are a total of 5,040 (=7!) evaluation patterns in case 1. The worst rating for the entire search was 0.014. These were the rating for mismatches sorted in ascending and descending order. On the other hand, the best rating for the entire search was 0.0008. The sequence of best ratings was 12 patterns. Fig. 5 shows all patterns when calculating according to the proposed algorithm. The best rating obtained with the proposed algorithm was 0.0008. In the top four patterns in Fig. 5, the calculation is interrupted in the middle. All other six patterns in the lower part of Fig. 5 equally obtain the best evaluation rating.

Case 2: Current mismatches ΔI/I -0.032, -0.024, -0.011, 0.001, 0.012, 0.021, 0.033

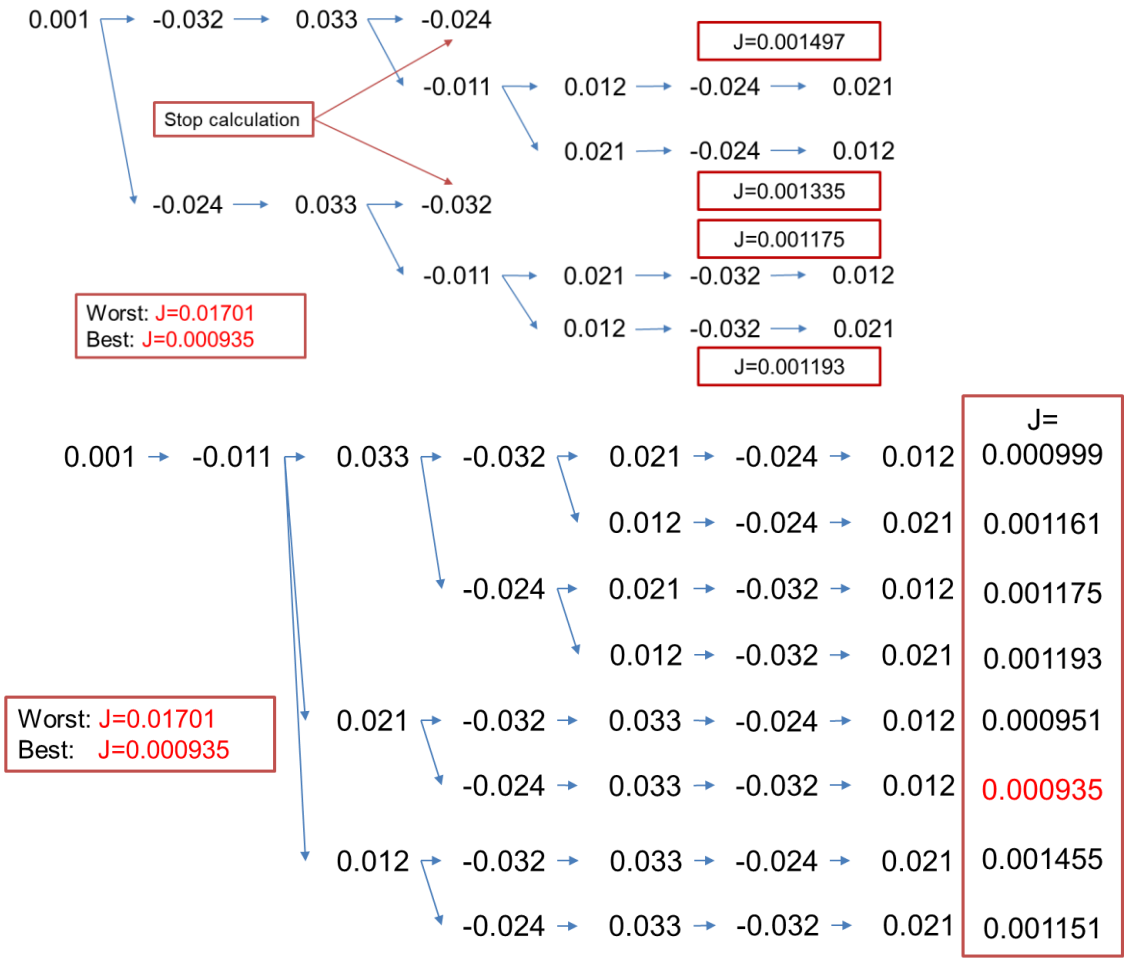

Fig. 6 Calculating process according to the algorithm in case 2.

There are a total of 5,040 (=7!) evaluation patterns in case 2. The worst rating for the entire search was 0.01701. These were the rating for mismatches sorted in ascending and descending order. On the other hand, the best rating for the entire search was 0.000935. The sequence of best ratings was 2 patterns. Fig. 6 shows all patterns when calculating according to the proposed algorithm. The best rating obtained with the proposed algorithm was 0.000935.

Case 3: Current mismatches ΔI/I -0.100, -0.085, -0.056, 0.022, 0.052, 0.077, 0.090

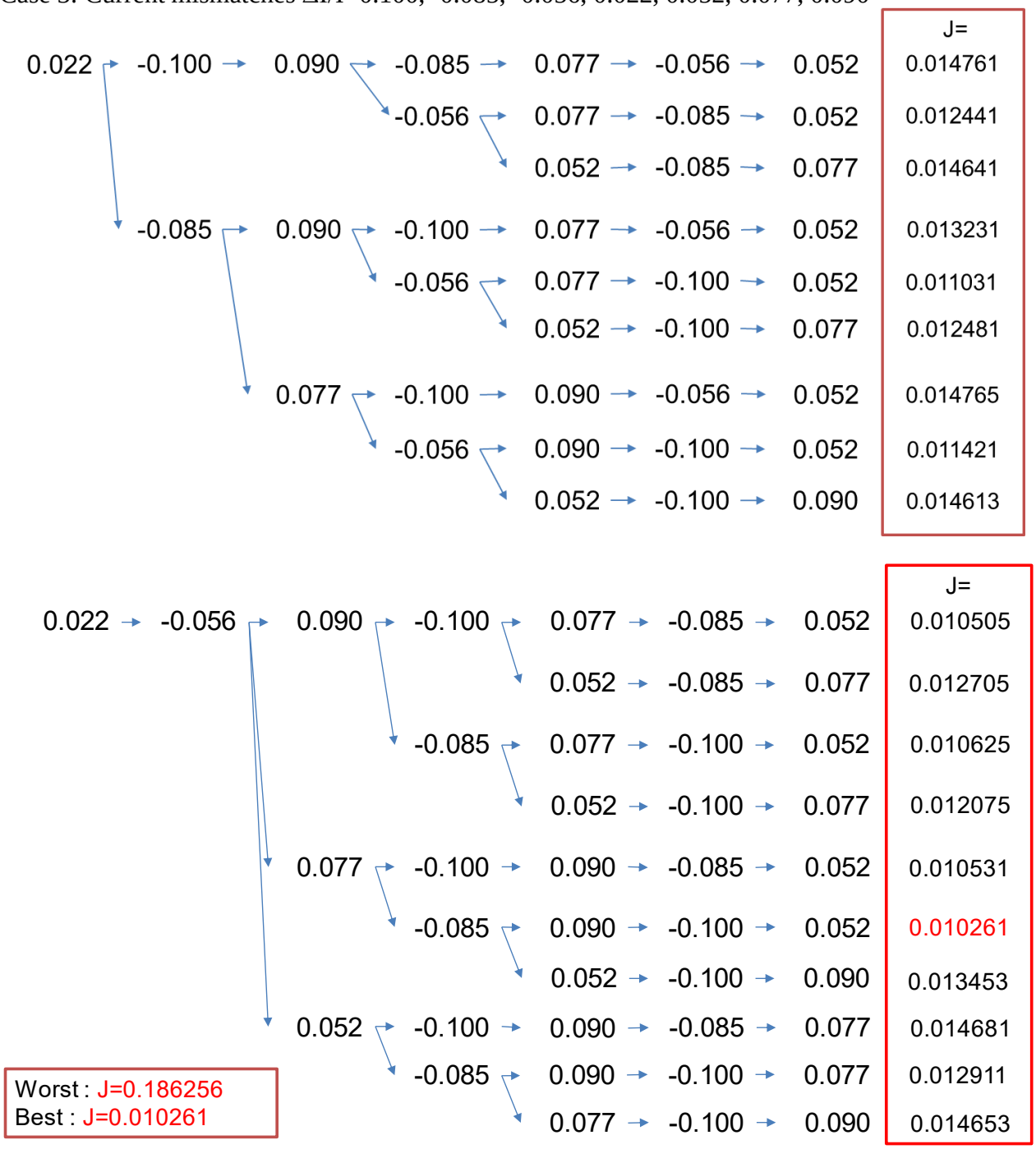

Fig. 7 Calculating process according to the algorithm in case 3.

There are a total of 5,040 (=7!) evaluation patterns in case 3. The worst rating for the entire search was 0.186256. These were the rating for mismatches sorted in ascending and descending order. On the other hand, the best rating for the entire search was 0.010261. The sequence of best ratings was 2 patterns. Fig. 7 shows all patterns when calculating according to the proposed algorithm. The best rating obtained with the proposed algorithm was 0.010261.

Case 4: Variation ΔI/I -0.030, -0.022, -0.017, 0.013, 0.013, 0.013, 0.030

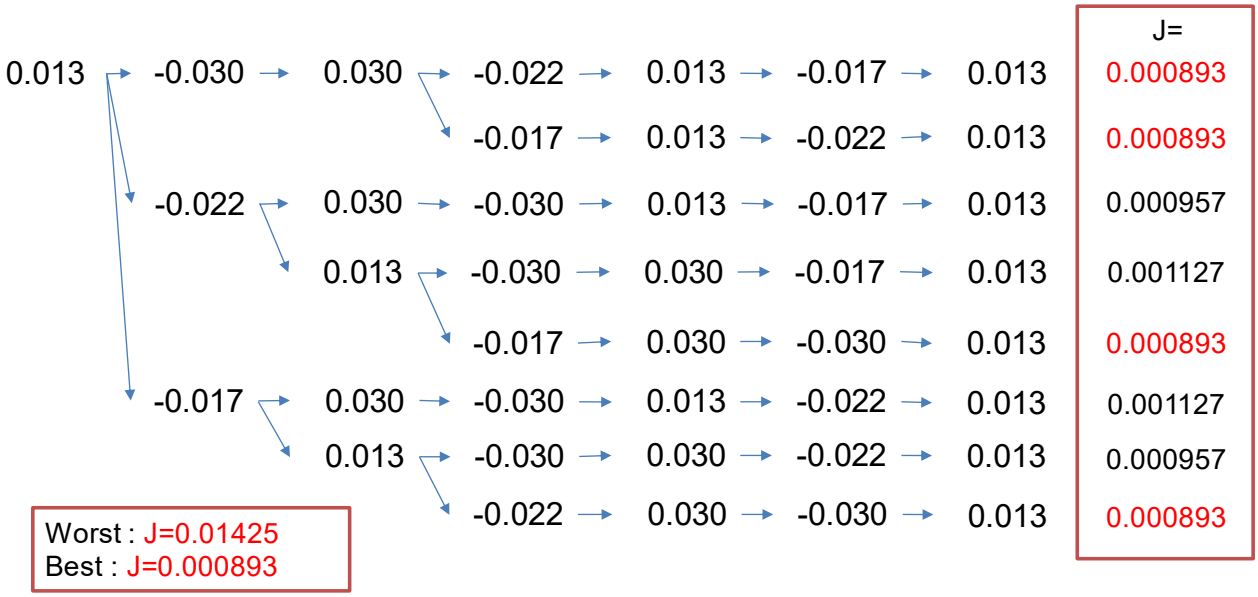

Fig. 8 Calculating process according to the algorithm in case 4.

Due to the duplicate current mismatch values, there are a total of 840 (=6!) evaluation patterns in case 4. The worst rating for the entire search was 0.01425. These were the rating for mismatches sorted in ascending and descending order. On the other hand, the best rating for the entire search was 0.000893. The sequence of best ratings was 4 patterns. Fig. 8 shows all patterns when calculating according to the proposed algorithm. The best rating obtained with the proposed algorithm was 0.000893.

Case 5: Current mismatches ΔI/I -0.070, -0.021, -0.002, -0.001, 0.001, 0.013, 0.080

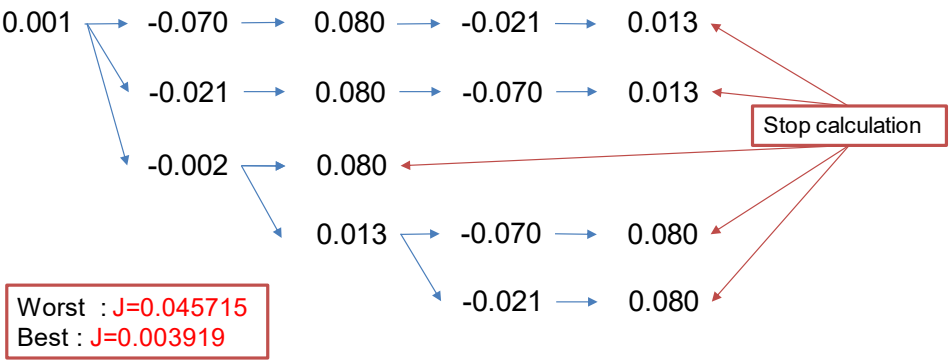

Fig. 9 Calculating process according to the algorithm in case 5.

There are a total of 5,040 (=7!) evaluation patterns in case 5. The worst rating for the entire search was 0.045715. These were the rating for mismatches sorted in ascending and descending order. On the other hand, the best rating for the entire search was 0.003919. The sequence of best ratings was 2 patterns. Fig. 9 shows all patterns when calculating according to the proposed algorithm. We have found that the calculation according to the proposed algorithm is interrupted in the middle for all possible patterns. In this case, the proposed algorithm does NOT work. Therefore, it was not possible to produce a J rating.

Case 6: Variation ΔI/I -0.070, -0.021, -0.017, 0.027, 0.027, 0.027, 0.027

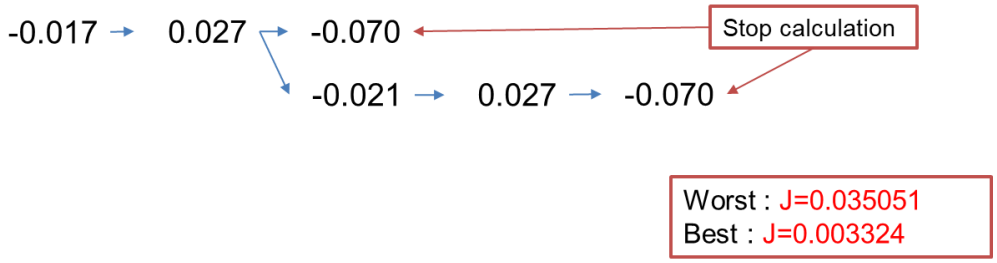

Fig. 10 Calculating process according to the algorithm in case 6.

Due to the duplicate permutations, there are a total of 210 (=7! / 4!) evaluation patterns in case 6. The worst rating for the entire search was 0.035051. These were the rating for mismatches sorted in ascending and descending order. On the other hand, the best rating for the entire search was 0.003324. The sequence of best ratings was 2 patterns. Fig. 10 shows all patterns when calculating according to the proposed algorithm. We have found that the calculation according to the proposed algorithm is interrupted in the middle for all possible patterns. In this case, the proposed algorithm does NOT work. Therefore, it was not possible to produce a J rating.

We see that in cases 1, 2, 3 and 4, the proposed algorithm work well while in cases 5 and 6, it does NOT work. Thus, the proposed algorithm is required to improve to work in any case. Therefore, instead of selecting values in sequence, try to exchange them from their original form. The flow of the algorithm is changed as follows:

1. From those in ascending order, replace the one closest to the mean with the first one.
2. Randomly replace the current average error with one that changes sign when added.
3. Repeat 2, while there is something that can change sign when added.
4. If there is nothing left to change the sign during the exchange, the exchange is interrupted and the evaluation is calculated.

The results of using this suboptimal algorithm for case 5 and case 6 are shown in Figs. 9-17.

Case 5: Current mismatches ΔI/I -0.070, -0.021, -0.002, -0.001, 0.001, 0.013, 0.080

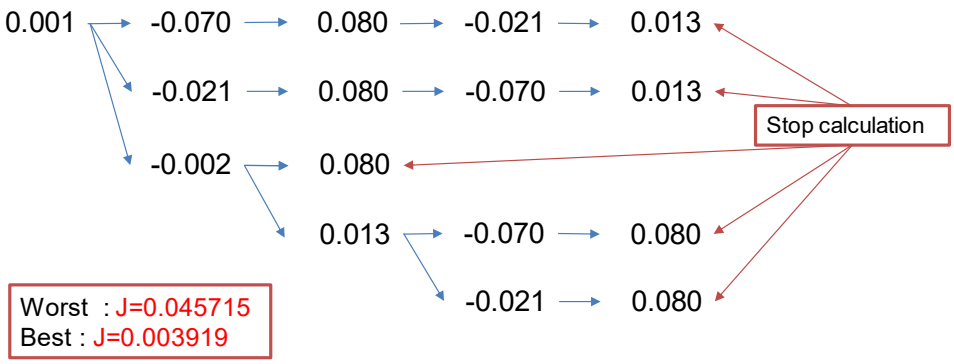

Fig. 9 Calculating process according to the algorithm in case 5.

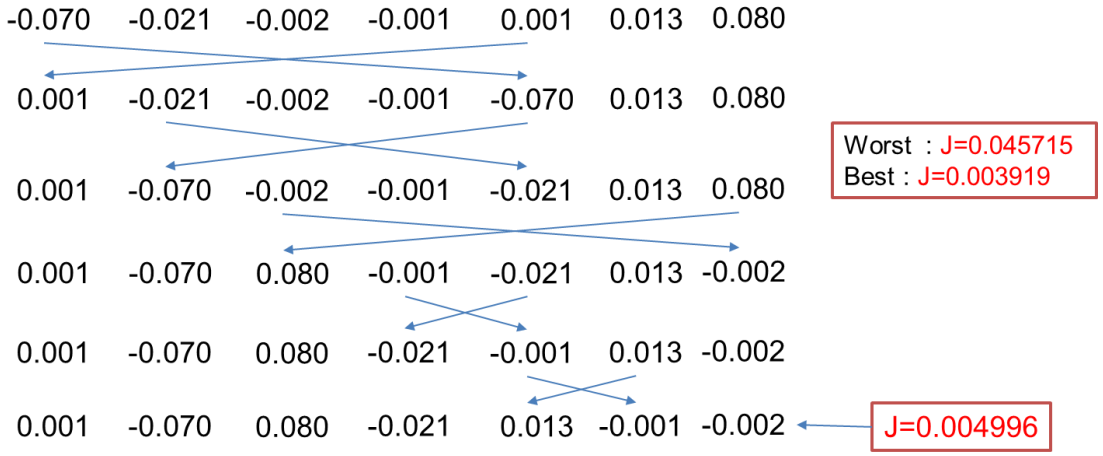

Fig. 11. 0.001, -0.070, 0.080, -0.021, 0.013 are exchanged in order in case 5

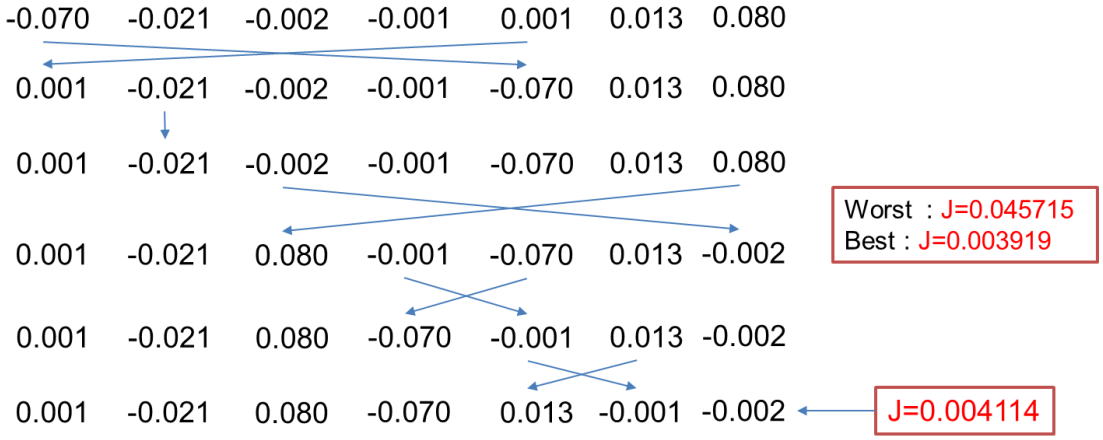

Fig. 12. 0.001, -0.021, 0.080, -0.070, 0.013 are exchanged in order in case 5

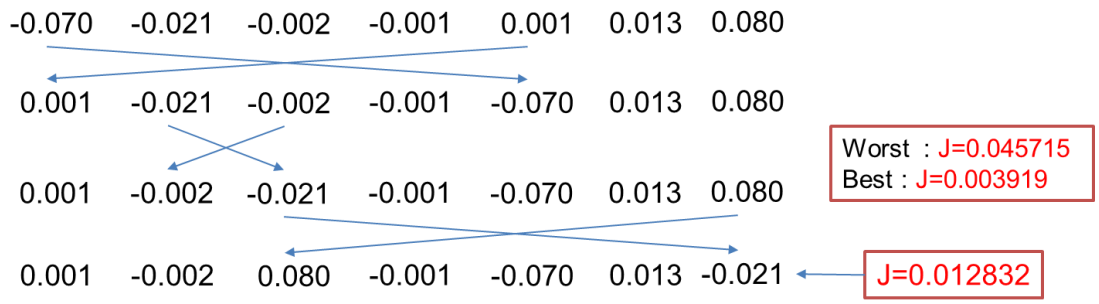

Fig. 13. 0.001, -0.002, 0.080, are exchanged in order in case 5

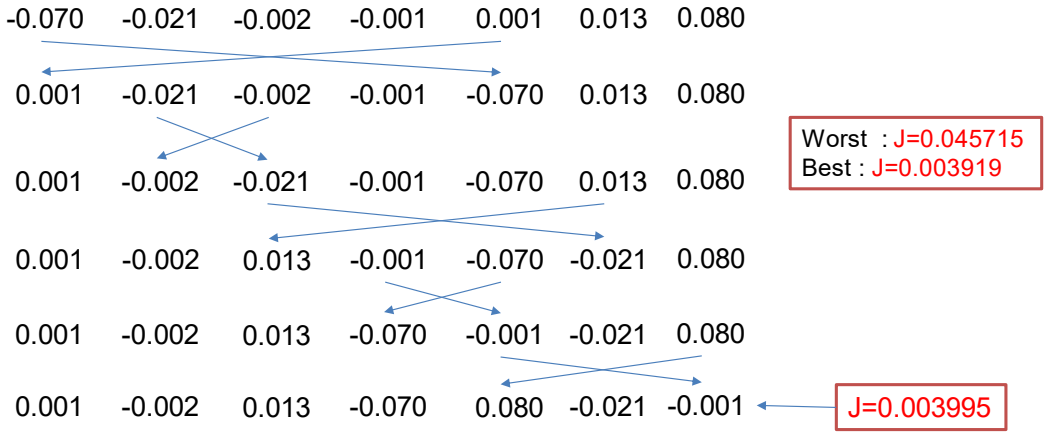

Fig. 14. 0.001, -0.002, 0.013, -0.070, 0.080 are exchanged in order in case 5

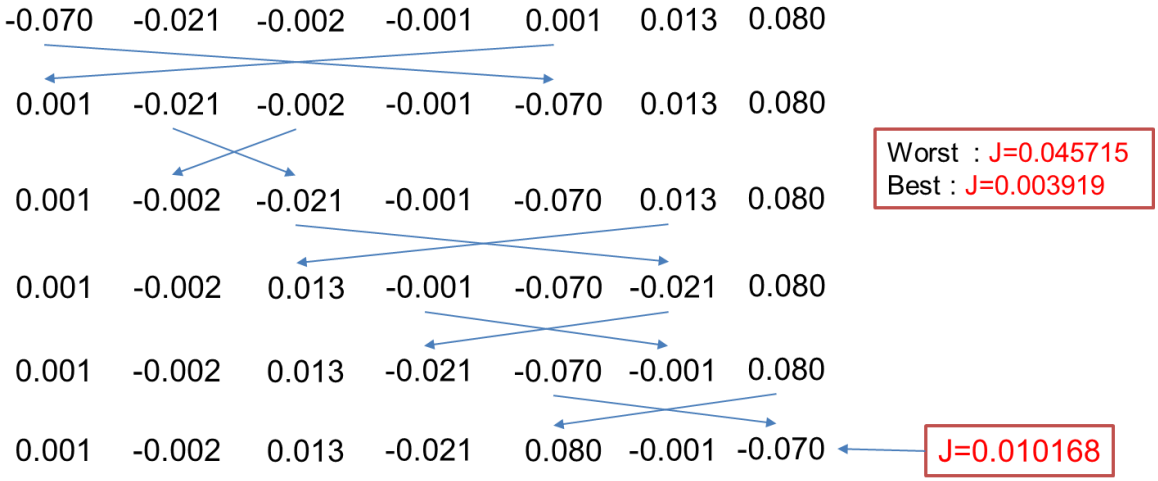

Fig. 15. 0.001, -0.002, 0.013, -0.021, 0.080 are exchanged in order in case 5

The suboptimal algorithm was used for the five patterns whose computation was interrupted in the middle in Fig. 9. The process is shown in Fig. 11. to Fig. 15. In case 5, the best rating was 0.003919 when the entire search was conducted. The best rating obtained by the suboptimal algorithm was 0.003995 in Fig. 14.

Case 6: Variation ΔI/I -0.070, -0.021, -0.017, 0.027, 0.027, 0.027, 0.027

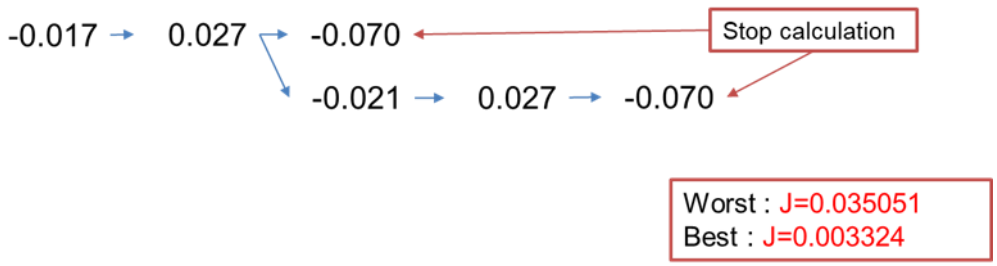

Fig. 10 Calculating process according to the algorithm in case 6.

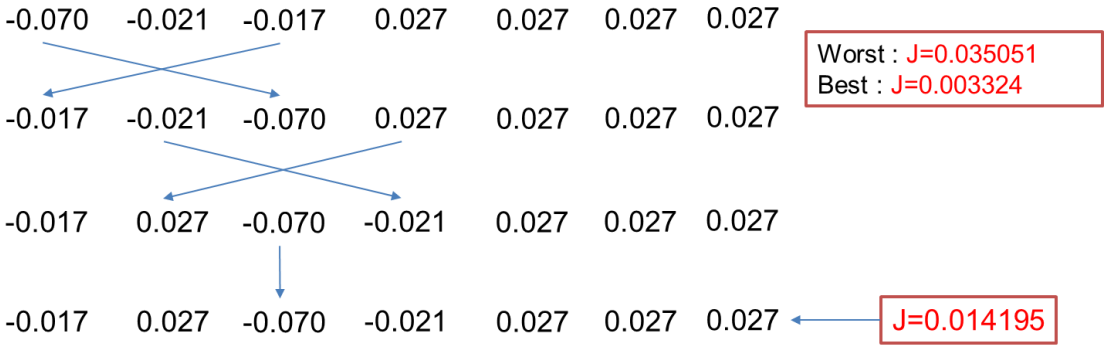

Fig. 16. -0.017, 0.027, -0.070 are exchanged in order in Case 6

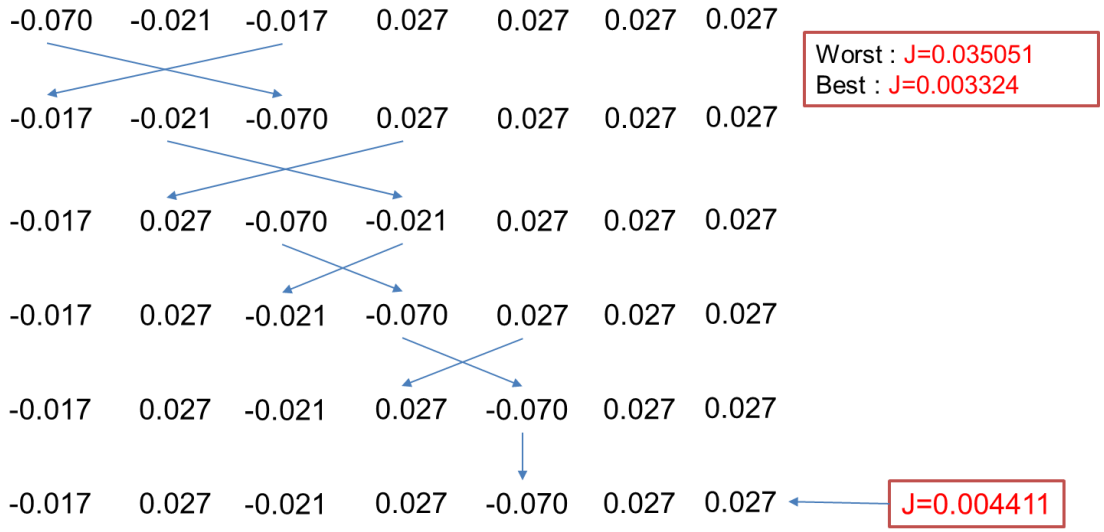

Fig. 17. -0.017, 0.027, -0.021, 0.027, -0.070 are exchanged in order in case 6

The suboptimal algorithm was used for the two patterns whose computation was interrupted in the middle in Fig. 10. The process is shown in Fig. 16. and Fig. 17. In case 6, the best rating was 0.003324 when the entire search was conducted. The best rating obtained by the suboptimal algorithm was 0.004411 in Fig. 17.

4. Conclusion

This paper describes the linearity self-calibration method for the high-resolution unary DAC with measured unit current values and their selection order arrangement for its INL reduction. Hence, we have investigated the reduction of number of their possible combinations with several methods for the calculating time reduction, and their simulation results have been shown for comparison. Then we have proposed an effective algorithm based on the consideration how the selection order can realize the DAC INL reduction, and shown its effectiveness with some examples.

Acknowledgements

The authors would like to thank Prof. Toru Sai for introducing his related research. Also, Jedat Inc. is acknowledged for their support of this research. Also, this work is supported by Adaptable and Seamless Technology Transfer Program through Target-Driven R&D (A-STEP) from Japan Science and Technology Agency (JST) Grant Number JPMJTR201C, and JSPS KAKENHI Grant Number 21K04190.

References

[1] F. Maloberti, Data Converters, ***Spring***, 2007.

[2] R. V. D. Plassche, Integrated Analog-to-Digital and Digital-to-Analog Converters, ***Springer***, 2012.

[3] M. J. M. Pelgrom, AAD C. J. Duinmaijer, A. O. G. Welbers, "Matching Properties of MOS Transistors", ***IEEE Journal of Solid-State Circuits,*** Vol. 24, No. 5, pp.1433-1440, Oct. 1989.

[4] Y. Liu, A. Kuwana, S. Katayama, X. Li, A. Motozawa, H. Kobayashi, "Segmented DAC Linearity Improvement Algorithm Using Unit Cell Sorted Alternately with Digital Method", ***24th Workshop on Synthesis and System Integration of Mixed Information Technologies,*** Hirosaki, Japan, Oct. 2022.

[5] Y. Liu, A. Kuwana, S. Katayama, X. Li, A. Motozawa, H. Kobayashi, "Optimization of Segmented DAC Linearity Improvement Algorithm Using Unit Cell Sorting with Digital Method", **T*he 31st International Workshop on Post-Binary ULSI Systems,*** May 2022.

[6] G. A. M. Van der Plas, J. Vandenbussche, W. Sansen, M. S. J. Steryaert, and G. G. E. Gielen, "A 14-bit Intrinsic Accuracy Q2 Random Walk CMOS DAC", ***IEEE Journal of Solid-State Circuits***, Vol. 34, No. 12, pp. 1708-1718, Dec. 1999.

[7] Y. Cong, R. L. Geiger, "Switching Sequence Optimization for Gradient Error Compensation in Thermometer-Decoded DAC Arrays", ***IEEE Trans. Circuits and Systems II***, Vo. 47, No. 7, pp. 585-595, July 2000.

[8] K.-C. Kuo, C. -W. Wu, "A Switching Sequence for Gradient Error Compensation in the DAC Design", ***IEEE Trans. Circuits and Systems II***, Vol. 58, No. 8, pp. 502-506, Aug 2011.

[9] X. Li, F. Qiao, H. Yang, "Balanced Switching Schemes for Gradient-Error Compensation in Current-Steering DACs", ***IEICE Trans. Electron***, Vol. E95-C, No.11, pp. 1790-1798, Nov. 2012.

[10] M. Higashino, S. N. B. Mohyar, H. Kobayashi, "DAC Linearity Improvement Algorithm with Unit Cell Sorting Based on Magic Square", ***IEEE International Symposium on VLSI Design, Automation and Test,*** Hsinchu, Taiwan, April 2016.

[11] D. Yao, Y. Sun, M. Higashino, S. N. Mohyar, T. Yanagida, T. Arafune, N. Tsukiji, H. Kobayashi, "DAC Linearity Improvement with Layout Technique Using Magic and Latin Squares", ***IEEE International Symposium on Intelligent Signal Processing and Communication Systems,*** Xiamen, China, Nov. 2017.

[12] M. Higashino, S. N. B. Mohyar, Y. Dan, Y. Sun, A. Kuwana, H. Kobayashi, "Digital-to-Analog Converter Layout Technique and Unit Cell Sorting Algorithm for Linearity Improvement Based on Magic Square", ***Journal of Technology and Social Science,*** Vol. 4, No. 1, pp. 22-35, Jan. 2020.

[13] M. S. Yenuchenk, A. S. Korotkov, D. V. Morozov, M. M. Pilipko, "A Switching Sequence for Unary Digital-to-Analog Converters Based on a Knight's Tour", ***IEEE Transactions on Circuits and Systems I***, Vol. 66, No. 6, pp. 2230-2239, June 2019.

[14] D. Yao, X. Bai, S. Katayama, A. Kuwana, K. Kawauchi, H. Kobayashi, K. Hirai, A. Suzuki, S. Yamada, T. Kato, R. Kitagawa, T. Shimamura, G. Adhikari, N. Ono, K. Miura, S. Yamaguchi, "Unit Cell Mismatch Scrambling Method for High-Resolution Unary DAC based on Virtual 3D Layout", ***IEICE Electronics Express***, Vol. 19, Issue 24, Pages 20220430, Dec. 2022.

[15] D. Yao, X. Bai, A. Kuwana, K. Kawauchi, Y. Zhao, J. Wei, S. Katayama, M. Higashino, H. Kobayashi, "Segmented DAC Unit Cell Selection Algorithm and Layout/Routing Based on Classical Mathematics", ***Journal of Mechanical and Electrical Intelligent System (JMEIS, J. Mech. Elect. Intel. Syst.),*** Vol.6, No.1, pp.13-30, Jan. 2023.