2015 International Symposium on Nonlinear Theory and its Applications
NOLTA2015, Kowloon, Hong Kong, China, December 1-4, 2015

Synthesis and analysis of piecewise rotational chaos for an optimizer

Yoshikazu YAMANAKA† and Tadashi TSUBONE‡

†‡Nagaoka University of Technology, 1603-1, Kamitomioka, Nagaoka, Niigata, 940-2188, Japan
†Email: y_yama@stn.nagaokaut.ac.jp, ‡Email: tsubone@vos.nagaokaut.ac.jp

Abstract—
Implement of optimization methods by the circuit is one of the practical ways to get the optimal solution within limited time. In previous study, we have proposed an optimizer based on chaotic dynamical system that seems to be suited for circuit implementation. In this paper, we consider the system parameter of the previous proposed method. A parameter set that obtains statistically better solution was discovered. Using the parameter, the chaotic system exhibits characteristic time-dependent behavior.

1. Introduction

Many problems in engineering field can be considered as optimization problems, such as design of artificial objects and arrangement of equipment. For the optimization problems, it is required to quickly get an optimal solution that is characterized by the maximum or minimum fitness value. Implementation of optimization methods by the circuit is one of the solutions of it.

Optimization methods that search the optima using some searching points based on dynamical systems would be feasible for the circuit implementation. This is because the updating rules of searching points can be written by different equations. However, most of them require stochastic terms. Implementation of the stochastic terms would make the circuit complex. In previous study, we have proposed an optimization method based on piecewise rotational chaotic system (OPRC) [1]. OPRC is deterministic system and the performance is better than particle swarm optimization [2]. It is expected that the circuit of OPRC is simpler than other dynamical-based optimization methods.

In this paper, we consider system parameters of OPRC that can obtain better solutions for general problems. Previously, parameters of OPRC are experimentally selected for each benchmark function. However, it is difficult to change the parameters in the circuit, comparing with numerical experiments. Hence, it is required to consider the adequate parameters that obtain better solutions for many functions. Our investigation discovered the adequate parameters. The correlation between the performance of OPRC and time-dependent behavior of the chaotic system was also observed.

In this paper, Sec. 2 describes the chaotic dynamical system and its time-dependent behavior. OPRC is mentioned in Sec. 3. Section 4 shows comparison results between some parameter sets of OPRC for benchmark functions.

2. A piecewise rotational chaotic system

2.1. Dynamics

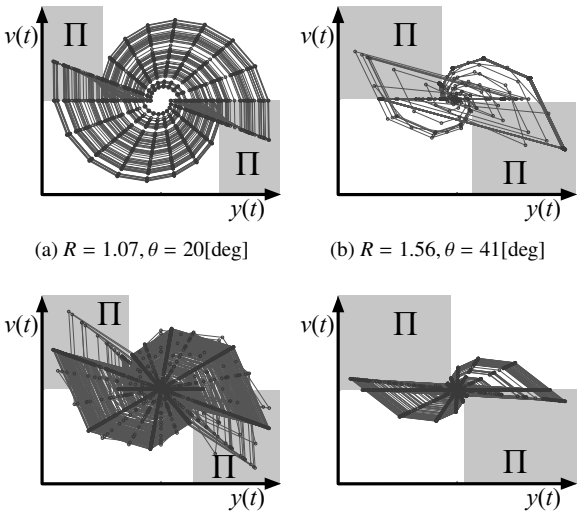
Dynamics is

$$\begin{bmatrix} y(t+1) \\ v(t+1) \end{bmatrix} = \begin{cases} \begin{bmatrix} 2\mathrm{sgn}(y(t))Th - y(t) \\ 0 \end{bmatrix} & \text{for}(v(t), y(t)) \in \Pi, \quad (1a) \\ R\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix} y(t) \\ v(t) \end{bmatrix} & \text{otherwise}, \quad (1b) \end{cases}$$

$$\Pi = \left\{ (v(t), y(t)) \;\middle|\; |y(t)| > Th, \mathrm{sgn}(v(t)y(t)) = -1 \right\},$$

$$\mathrm{sgn}(a) = \begin{cases} 1 & \text{for } a > 0, \\ -1 & \text{for } a \le 0. \end{cases}$$

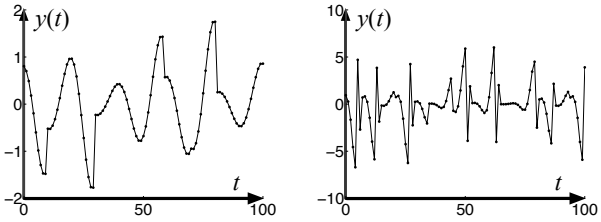
where $R > 1$, $0 < \theta < \frac{\pi}{2}$ and Th are system parameters, R and θ refers to the expanding and rotational motion, respectively. Th denotes $y(t)$-axis threshold of Π. The trajectory rotates divergently around the origin by Eq. (1b), and when the trajectory gets into Π, it is folded forward to the origin by Eq.(1a). The trajectory repeats this manner and exhibits chaotic attractor as show in Fig. 1. Figure 2 shows time-series of the $y(t)$.



(a) $R = 1.07, \theta = 20$[deg] (b) $R = 1.56, \theta = 41$[deg]

(c) $R = 1.37, \theta = 50$[deg] (d) $R = 1.69, \theta = 31$[deg]

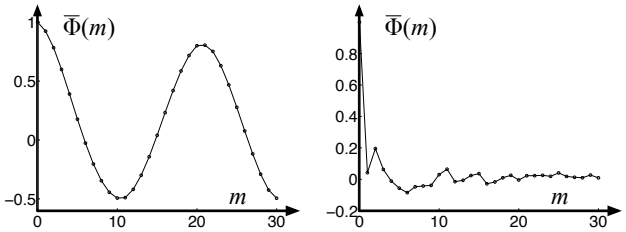
Figure 1: Typical chaotic attractors of PRC



(a) $R = 1.07, \theta = 20$[deg] (b) $R = 1.56, \theta = 41$[deg]

(c) $R = 1.37, \theta = 50$[deg] (d) $R = 1.69, \theta = 31$[deg]

Figure 2: Typical time-series of $y(t)$



(a) $R = 1.07, \theta = 20$[deg] (b) $R = 1.56, \theta = 41$[deg]

(c) $R = 1.37, \theta = 50$[deg] (d) $R = 1.69, \theta = 31$[deg]

Figure 3: Autocorrelation function of $y(t)$

2.2. Time-series analysis

Let us consider the time-dependent behavior of PRC's trajectory. We calculated autocorrelation function (ACF) of $y(t)$ that is given by Eq. (1). We gave $y_i(0)$ and $v_i(0)$, where $i \in \{1 \dots 100\}$, by uniformed distribution with $[-1, 1]$ and $Th = 1$, then update $y_i(t)$ and $v_i(t)$ until $t = 11000$. Focusing on the time-series of $y_i(t)$, the ACF $\phi_i(m)$ and the normalized ACF $\Phi_i(m)$ is calculated by

$$\phi_i(m) = \sum_{n=10000}^{11000-m-1} y_i(n) y_i(n+m), \tag{2}$$

$$\Phi_i(m) = \frac{\phi_i(m)}{\phi_i(0)}. \tag{3}$$

The ACF averaged by 100 sets of different initial conditions, $\overline{\Phi}(m)$, is given by

$$\overline{\Phi}(m) = \frac{1}{100} \sum_{i=1}^{100} \Phi_i(m). \tag{4}$$

Figure 3 shows typical $\overline{\Phi}(m)$. $y(t)$ given by $R = 1.07, \theta = 20$[deg] exhibits slowly damping and low-frequency $\overline{\Phi}(m)$ as shown in Fig. 3a. The ACF shows that $y(t)$ keeps strong time-correlation at least $m = 30$. In Fig. 3b, $\overline{\Phi}(m)$ seems converging faster than other typical parameters. It indicates that the time-series given by $R = 1.56, \theta = 41$[deg] has weak time-correlation. $y(t)$ given by $R = 1.37, \theta = 50$[deg] obtains damped oscillated $\overline{\Phi}(m)$, and it converges on almost 0 at $m = 30$ as shown in Fig. 3c. Damped oscillated $\overline{\Phi}(m)$ is also observed with $y(t)$ given by $R = 1.69, \theta = 31$[deg] as shown in Fig. 3d. Comparing Fig. 3c, the ACF has more high-frequency and converges faster. The $y(t)$ has time-correlation for only a short time.

3. An optimization method based on PRC

This section describes OPRC [1]. OPRC searches the optimal solution by n number of searching points

$$\boldsymbol{x}_i(t) = \{x_{i1}, x_{i2}, \dots, x_{id}\}$$

and independent variables

$$\boldsymbol{v}_i(t) = \{v_{i1}, v_{i2}, \dots, v_{id}\}$$

where $i \in \{1, 2, \dots n\}$ is an index number, d is a number of dimension of a given problem f, and $t \in \{0, 1, 2, \dots\}$ is time step. $\boldsymbol{v}_i(0)$ and $\boldsymbol{x}_i(0)$ are set to 0 and given by randomly, respectability. The fitness value $f(\boldsymbol{x}_i(t))$ is calculated for each time step. $\boldsymbol{x}_i(t)$ is stored as $\boldsymbol{pb}_i$ when $f(\boldsymbol{x}_i(t))$ is the maximum or minimum value in own fitness history, $\{f(\boldsymbol{x}_i(0)), f(\boldsymbol{x}_i(1)), f(\boldsymbol{x}_i(2)), \dots\}$. $\boldsymbol{pb}_i$ is stored as $\boldsymbol{gb}$ when $f(\boldsymbol{pb}_i)$ is the best in $\{f(\boldsymbol{pb}_1), f(\boldsymbol{pb}_2), \dots, f(\boldsymbol{pb}_n)\}$. To update $\boldsymbol{x}_i(t)$ with considering the $\boldsymbol{pb}_i$ and $\boldsymbol{gb}$, let shifted position of $\boldsymbol{x}_i(t)$ be $\boldsymbol{y}_i(t)$:

$$\boldsymbol{y}_i(t) = \boldsymbol{x}_i(t) - \frac{1}{2}(\boldsymbol{pb}_i + \boldsymbol{gb}). \tag{5}$$

When $\boldsymbol{y}_i(t) = 0$, $\boldsymbol{x}_i(t)$ is the middle point between $\boldsymbol{pb}_i$ and $\boldsymbol{gb}$. $\boldsymbol{y}_i(t)$ and $\boldsymbol{v}_i(t)$ are updated by PRC dynamics as follows:

$$\begin{bmatrix} y_{ij}(t+1) \\ v_{ij}(t+1) \end{bmatrix} = \begin{cases} \begin{bmatrix} 2\mathrm{sgn}(y_{ij}(t)) Th_{ij} - y_{ij}(t) \\ 0 \end{bmatrix} & \text{for} (v_{ij}(t), y_{ij}(t)) \in \Pi, \quad \text{(6a)} \\ R \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} y_{ij}(t) \\ v_{ij}(t) \end{bmatrix} & \text{otherwise,} \quad \text{(6b)} \end{cases}$$

$$\Pi = \left\{ (v_{ij}(t), y_{ij}(t)) \middle| |y_{ij}(t)| > Th_{ij}, \mathrm{sgn}(v_{ij}(t) y_{ij}(t)) = -1 \right\},$$

where $j \in \{1, 2, \ldots, d\}$ is an index number, and $Th_{ij} = \frac{1}{2}|pb_{ij} - gb_j|$. The next searching points are given by

$$\boldsymbol{x}_i(t+1) = \boldsymbol{y}_i(t+1) + \frac{1}{2}(\boldsymbol{pb}_i + \boldsymbol{gb}).$$

The searching points are updated until the termination criterion, $t_{\max}$, has been met. At the criterion, $\boldsymbol{gb}$ is a searched solution.

4. Numerical experiment

Table 1: CEC 2013 benchmark functions [3].

Type	f	Optimal fitness	Search range
Unimodal	f_1	-1400	$[-100, 100]^D$
Unimodal	f_2	-1300	$[-100, 100]^D$
Unimodal	f_3	-1200	$[-100, 100]^D$
Unimodal	f_4	-1100	$[-100, 100]^D$
Unimodal	f_5	-1000	$[-100, 100]^D$
Multimodal	f_6	-900	$[-100, 100]^D$
Multimodal	f_7	-800	$[-100, 100]^D$
Multimodal	f_8	-700	$[-100, 100]^D$
Multimodal	f_9	-600	$[-100, 100]^D$
Multimodal	f_{10}	-500	$[-100, 100]^D$
Multimodal	f_{11}	-400	$[-100, 100]^D$
Multimodal	f_{12}	-300	$[-100, 100]^D$
Multimodal	f_{13}	-200	$[-100, 100]^D$
Multimodal	f_{14}	-100	$[-100, 100]^D$
Multimodal	f_{15}	100	$[-100, 100]^D$
Multimodal	f_{17}	300	$[-100, 100]^D$
Multimodal	f_{18}	400	$[-100, 100]^D$
Multimodal	f_{19}	500	$[-100, 100]^D$
Multimodal	f_{20}	600	$[-100, 100]^D$
Composition	f_{21}	700	$[-100, 100]^D$
Composition	f_{22}	800	$[-100, 100]^D$
Composition	f_{23}	900	$[-100, 100]^D$
Composition	f_{24}	1000	$[-100, 100]^D$
Composition	f_{25}	1100	$[-100, 100]^D$
Composition	f_{26}	1200	$[-100, 100]^D$
Composition	f_{27}	1300	$[-100, 100]^D$
Composition	f_{28}	1400	$[-100, 100]^D$

Table 2: Parameters

	R	θ[deg]
Set 1	1.07	20
Set 2	1.56	41
Set 3	1.37	50
Set 4	1.69	31

Table 3: Benchmark results

Gray rectangles denote the best averaged value for each function.

f		set 1	set 2	set 3	set 4
f_1	ave	-1.40×10^3	-1.40×10^3	-1.40×10^3	-1.40×10^3
	std	2.25×10^{-13}	1.69×10^{-7}	3.25×10^{-14}	8.59×10^{-2}
f_2	ave	2.31×10^6	3.04×10^6	1.84×10^5	4.16×10^6
	std	1.85×10^6	2.42×10^6	1.75×10^5	3.01×10^6
f_3	ave	3.23×10^8	4.95×10^7	7.83×10^7	9.78×10^8
	std	5.30×10^8	9.82×10^7	2.03×10^8	2.34×10^9
f_4	ave	2.55×10^4	3.56×10^4	1.79×10^4	4.67×10^4
	std	1.25×10^4	2.59×10^4	1.32×10^4	6.39×10^4
f_5	ave	-1.003	-1.00×10^3	-1.00×10^3	-1.00×10^3
	std	3.34×10^{-8}	6.08×10^{-5}	8.59×10^{-14}	3.19×10^{-1}
f_6	ave	-8.832	-8.92×10^2	-8.97×10^2	-8.85×10^2
	std	2.35×10^1	1.01×10^1	1.03×10^1	2.18×10^1
f_7	ave	-7.422	-7.61×10^2	-7.58×10^2	-7.39×10^2
	std	3.48×10^1	2.15×10^1	2.86×10^1	3.28×10^1
f_8	ave	-6.802	-6.80×10^2	-6.80×10^2	-6.80×10^2
	std	9.33×10^{-2}	7.97×10^{-2}	9.35×10^{-2}	1.15×10^{-1}
f_9	ave	-5.942	-5.91×10^2	-5.94×10^2	-5.91×10^2
	std	1.68	1.58	1.53	1.40
f_{10}	ave	-4.952	-4.99×10^2	-5.00×10^2	-4.95×10^2
	std	4.52	3.64×10^{-1}	2.84×10^{-1}	4.76
f_{11}	ave	-3.802	-3.95×10^2	-3.88×10^2	-3.88×10^2
	std	9.27	2.23	7.77	5.17
f_{12}	ave	-2.742	-2.68×10^2	-2.78×10^2	-2.71×10^2
	std	1.28×10^1	8.07	1.05×10^1	1.27×10^1
f_{13}	ave	-1.532	-1.66×10^2	-1.63×10^2	-1.62×10^2
	std	1.94×10^1	1.03×10^1	1.82×10^1	1.45×10^1
f_{14}	ave	5.53×10^2	7.51×10^2	3.15×10^2	7.05×10^2
	std	2.11×10^2	3.08×10^2	1.89×10^2	3.92×10^2
f_{15}	ave	9.83×10^2	1.84×10^3	1.04×10^3	1.62×10^3
	std	3.15×10^2	2.50×10^2	3.11×10^2	4.82×10^2
f_{17}	ave	3.29×10^2	3.27×10^2	3.19×10^2	3.31×10^2
	std	8.68	5.45	4.20	8.73
f_{18}	ave	4.32×10^2	4.47×10^2	4.30×10^2	4.51×10^2
	std	1.07×10^1	6.91	1.15×10^1	1.05×10^1
f_{19}	ave	5.01×10^2	5.02×10^2	5.01×10^2	5.02×10^2
	std	5.63×10^{-1}	4.35×10^{-1}	3.63×10^{-1}	6.60×10^{-1}
f_{20}	ave	6.04×10^2	6.04×10^2	6.04×10^2	6.04×10^2
	std	5.29×10^{-1}	3.43×10^{-1}	3.83×10^{-1}	3.64×10^{-1}
f_{21}	ave	1.09×10^3	1.09×10^3	1.09×10^3	1.10×10^3
	std	4.96×10^1	3.96×10^1	3.41×10^1	1.37×10^1
f_{22}	ave	1.65×10^3	1.75×10^3	1.34×10^3	1.79×10^3
	std	2.83×10^2	3.75×10^2	2.17×10^2	5.17×10^2
f_{23}	ave	2.14×10^3	2.84×10^3	2.08×10^3	2.66×10^3
	std	3.03×10^2	2.17×10^2	3.70×10^2	4.33×10^2
f_{24}	ave	1.21×10^3	1.23×10^3	1.22×10^3	1.23×10^3
	std	1.47×10^1	2.29	4.75	3.99
f_{25}	ave	1.31×10^3	1.33×10^3	1.31×10^3	1.33×10^3
	std	1.32×10^1	1.50	4.59	2.27
f_{26}	ave	1.44×10^3	1.44×10^3	1.44×10^3	1.45×10^3
	std	8.24×10^1	6.68×10^1	8.07×10^1	7.05×10^1
f_{27}	ave	1.91×10^3	1.76×10^3	1.76×10^3	1.83×10^3
	std	1.12×10^2	4.84×10^1	3.65×10^1	6.43×10^1
f_{28}	ave	2.34×10^3	1.85×10^3	1.96×10^3	1.82×10^3
	std	2.36×10^2	2.70×10^2	2.40×10^2	1.98×10^2

Table 4: Rank_{ij} based on averaged values in Table 3.

f	set 1	set 2	set 3	set 4
f_1	1.5	3	1.5	4
f_2	2	3	1	4
f_3	3	1	2	4
f_4	2	3	1	4
f_5	2	3	1	4
f_6	4	2	1	3
f_7	3	1	2	4
f_8	2	4	1	3
f_9	2	3	1	4
f_{10}	3	2	1	4
f_{11}	4	1	2	3
f_{12}	2	4	1	3
f_{13}	4	1	2	3
f_{14}	2	4	1	3
f_{15}	1	4	2	3
f_{17}	3	2	1	4
f_{18}	2	3	1	4
f_{19}	2	3	1	4
f_{20}	2	3	1	4
f_{21}	1	3	2	4
f_{22}	2	3	1	4
f_{23}	2	4	1	3
f_{24}	1	3	2	4
f_{25}	1	3	2	4
f_{26}	2	3	1	4
f_{27}	4	2	1	3
f_{28}	4	2	3	1
Rank_j	63.5	73	37.5	96

This section compares the performance of OPRC for benchmark functions. We selected 27 functions from CEC 2013 benchmark test suite [3] as shown in Table 1. We selected 4 typical parameters as shown in Table 2. The behavior of PRC using these parameters is described in previous sections. The conditions of the comparison are as follows: number of searching points is 20, size of dimension is 10, $t_{\max} = 1000$. For each trial, the initial values of searching points are given by the uniformed distribution with the range in Table 1. The fitness of $\boldsymbol{gb}$ at $t_{\max}$ is stored for each trial, and the final fitness values are averaged by 50 trials. Table 3 shows the averaged fitness values and the standard deviations. OPRC with parameter set 3 obtained the best-averaged values for 18 functions, and it seems superior to others. We tested the difference between the parameter sets with pairwise-test [4]. The rank of i-th function and j-th parameter set, Rank_{ij}, is shown in Table 4. Two parameter sets are significantly different when following an inequality is satisfied:

$$|\mathrm{Rank}_n - \mathrm{Rank}_m| > t_{1-0.01/2}\sqrt{\frac{2b(A-B)}{(b-1)(k-1)}} = 18.3, \quad (7)$$

where Rank_n and Rank_m is sum of ranks calculated by $\mathrm{Rank}_j = \sum_{i=1}^{b} \mathrm{Rank}_{ij}$, n and m are indexes of a compared parameter set, $t_{1-0.01/2}$ is quantile of the F-distribution with $k_1 = k - 1$ and $k_2 = (b-1)(k-1)$ degrees of freedom at 0.01 significant level, b and k is a number of functions and parameter sets, respectively, $A = \sum_{i=1}^{b}\sum_{j=1}^{k}\left(\mathrm{Rank}_{ij}\right)^2$, and $B = \frac{1}{b}\sum_{j=1}^{k}\left(\mathrm{Rank}_j\right)^2$. The difference between Rank_3 and other sum of ranks is over 18.3, therefore, it is considered that the parameter set 3 outperforms than others.

As shown in Sec. 2, the various time-dependent behavior of PRC is observed among the 4 parameters sets. And performance of OPRC is significantly different depending on the parameters. This comparison indicates that there is correlation between time-dependent behavior of PRC and performance of OPRC.

5. Conclusion

Performances of OPRC between typical parameter sets are compared statistically. A parameter set that obtains better result than others are discovered. Moreover, in this comparison performance of OPRC seems to be correlated with time-depend behavior of PRC. The analysis of the correlation and the circuit implementation of PRC is future works. We are working on the implementation by field programmable analog arrays.

References

[1] Y. Yamanaka and T. Tsubone, "A Basic Study on Symmetrical Chaotic Dynamics for Population-based Optimization," *Proceedings of the 2014 International Symposium on Nonlinear Theory and its Applications*, no. 5, pp. 152–155, September 2014.

[2] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of IEEE international conference on neural networks*, vol. 4, pp. 1942–1948, 1995.

[3] J. J. Liang, B. Y. Qu, P. N. Suganthan, and A. G. Hernández-Díaz, "Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization," *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report*, vol. 201212, 2013.

[4] W. J. Conover, *Practical Nonparametric Statistics*, vol. 40. Wiley, 2nd ed., December 1980.