Invited Paper

Bifurcation analysis by particle swarm optimization

Haruna Matsushita^{1a)}, *Hiroaki Kurokawa*², and
*Takuji Kousaka*³

¹ *Dept. of Electronics and Information Engineering, Kagawa University,
2217-20 Hayashi-cho, Takamatsu, Kagawa 761-0396, Japan*

² *Dept. of Electric and Electronic Engineering, Tokyo University of Technology,
1404-1 Katakuramachi, Hachioji City, Tokyo 192-0982, Japan*

³ *Dept. of Electrical and Electronic Engineering, Chukyo University,
101-2 Yagoto Honmachi, Showa-ku, Nagoya-shi, 466-8666, Japan*

^{a)} *haruna@ieee.org*

Received January 31, 2020; Revised May 13, 2020; Published October 1, 2020

Abstract: This paper explains a bifurcation parameter detection strategy based on particle swarm optimization (PSO), and it shows application examples on detection of local bifurcation parameters appearing in discrete-time dynamical systems and non-autonomous continuous dynamical systems. The algorithm design and analysis part shows that the PSO-based algorithm is fairly simple, easily understandable, and easily implementable. The method requires no careful initialization, exact calculation, gradient information of the system, or Lyapunov exponents. However, the simulation results show that the method accurately detects the local bifurcation parameters regardless of the stability of the periodic point.

Key Words: bifurcation point detection, bifurcation diagram, initial setup problem, particle swarm optimization (PSO)

1. Introduction

Complex real-world dynamics or phenomena depend on time and can be modeled as dynamical systems. A system described by ordinary differential equations (ODEs) is said to be continuous dynamical systems, and a system described by difference equations is said to be a discrete-time dynamical system or a map. Furthermore, by using switching, and/or combining differential equations and maps, various real-world systems can be described by mathematical formulas as the dynamical systems. Thus, analyzing such dynamical systems is important to solve a variety of problems in the real world, and dynamical systems research covers a wide range, embracing engineering, biology, sociology, ecology, and other fields. Such dynamical systems mostly contain one or more parameters. In general, the behavior is nonlinear and bifurcation phenomena can be observed. The *bifurcation* occurs when a small change, applied to the parameter values of a system (referring to bifurcation parameter), causes a sudden qualitative change in the periodic solutions' behavior. Bifurcation analysis, which investigates

how bifurcations depend on the system parameters, is one of the most important nonlinear analysis techniques for understanding phenomena of systems since small changes in the parameter values may have large effects on the system behaviors.

A bifurcation diagram, which is a graph of bifurcation sets, is often used to visualize a bifurcation structure. The simplest method for providing a two-parameter bifurcation diagram is a brute-force method. However, since an accuracy of the brute-force bifurcation diagram depends on the number of points on the grid in target parameter space, higher accuracy bifurcation diagram requires larger amount of calculations. Furthermore, due to its character, the brute-force method cannot detect unstable periodic point which is dynamically unstable. Meanwhile, various detection and tracing methods have been developed for bifurcation parameters [1–4], and a number of software packages are available for bifurcation analysis, such as AUTO [5] (and its specialized drivers such as SLIDE-CONT [6]) and BunKi [7]. However, these standard methods are based on the Newton–Raphson method, which is a gradient-based algorithm that requires the derivative of a function and appropriate initial conditions. Thus, such methods can be difficult for beginners who are unfamiliar with nonlinear analysis.

Alternatively, bifurcation analysis can be performed by population-based optimization (also called swarm intelligence) which is a metaheuristic algorithm with multiple potential solutions. Most population-based optimizations avoid the derivative of the objective functions, and a particularly popular algorithm is Particle Swarm Optimization (PSO) [8]. Despite its simplicity, the PSO performs global search and has many real/potential applications. Our research group proposed a PSO-based method to detect bifurcation parameters in discrete-time and non-autonomous dynamical systems [9–11]. The method, called nested layer particle swarm optimization (NLPSO), is a non-Newton method and is performed by two nested PSOs. Although the NLPSO requires no strict settings of initial values or cumbersome manual calculations, it can accurately and directly find the bifurcation parameter set without a Lyapunov or gradient-based method, regardless of the periodic point stability. The initial value problem of Newton-based bifurcation analyses can then be solved by assigning the NLPSO-detected single bifurcation parameter set as the initial point of the bifurcation curve tracing. Furthermore, NLPSO can also obtain a bifurcation diagram of both unstable and stable periodic points by plotting multiple bifurcation parameter sets detected by repeating the implementation of the NLPSO with different random initial states.

This paper explains the NLPSO-based detection strategy of the bifurcation parameter sets appearing in discrete-time dynamical systems and non-autonomous continuous dynamical systems. The paper is written with the needs of an expert at the dynamical systems in mind, however, those less familiar with the bifurcation analysis or the swarm intelligence should also be able to understand it.

The rest of this paper is organized as follows. Section 2 provides the PSO algorithm. Section 3 defines dynamical systems and the bifurcation of its periodic solutions. Section 4 describes the structure and algorithm of the NLPSO by incorporating two objective functions. Section 5 establishes the validity and applicability of NLPSO-based method using several examples. Section 6 provides conclusions and identifies several directions for future research.

2. Particle swarm optimization (PSO)

The PSO is one of the simplest and most popular population-based optimization techniques that exploits a population of individuals that correspond to multiple potential solutions. The population is called *swarm*, and the individuals (i.e., the objective variables of the optimization) are called *particles*. PSO searches the optimal solution by moving the particles around in the search space according to simple updating equations. Each particle moves with being influenced by its personal best position and the swarm best position. This has an effect of converging the swarm to a better solution.

PSO is a metaheuristic optimization method since it makes few or no assumptions about the optimization problem being solved. However, PSO does not guarantee that a global optimum can be found, and the solution found depends on the set of random variables generated because it includes some stochastic factors. Meanwhile, although PSO avoids the derivative of the optimization problem

Table I. Variables and their symbols of two PSOs in NLPSO.

Variable	Symbol		
	PSO	PSO _{bif}	PSO _{pp}
Particle index		i	
Index of $gbest$		g	
Dimension index		d	
Iteration counter	h	h_1	h_2
Position vector	$\mathbf{z}_i$	$\mathbf{z}_{bi}$	$\mathbf{z}_{pi}$
Velocity vector	$\mathbf{v}_i$	$\mathbf{v}_{bi}$	$\mathbf{v}_{pi}$
$pbest$ vector	$\mathbf{p}_i$	$\mathbf{p}_{bi}$	$\mathbf{p}_{pi}$
$gbest$ vector	$\mathbf{g}$	$\mathbf{g}_b$	$\mathbf{g}_p$
Number of particles	M	M_{bif}	M_{pp}
Number of search space dimensions	D	L	N
Objective function	$F(\cdot)$	$F_{bif}(\cdot)$	$F_{pp}(\cdot)$
Maximum iterations	H	H_1	H_2
Stop criteria	C	C_{bif}	C_{pp}

(i.e. the objective function of the optimization) as is required by gradient-based algorithms such as Newton method and gradient descent, and it can find good solutions by searching over very large spaces of potential solutions. These characteristics mean that PSO is applicable to discontinuous objective function with switch or jump, and it can find the optimal solution without appropriate initial conditions for the optimization.

The PSO algorithm is explained as follows. Each particle carries position and velocity information. Suppose a D -dimensional search space $\mathcal{S}$ and a swarm contains M particles, the position and velocity vectors of the i -th particle at iteration h are represented by $\mathbf{z}_i(h) = (z_{i1}, z_{i2}, \dots, z_{iD}) \in \mathbb{R}^D$ and $\mathbf{v}_i(h) = (v_{i1}, v_{i2}, \dots, v_{iD}) \in \mathbb{R}^D$, where $i = 1, 2, \dots, M$. Table I summarizes the symbols used in PSO.

Initially, at $h = 0$, the PSO particles are randomly distributed in the search space $[z_{\min}, z_{\max}]^D$, where $z_{\min}$ and $z_{\max}$ represent the lower and upper boundaries of the search-space $\mathcal{S}$. The velocities are initialized to zero to prevent particles from leaving the search space during the first iterations. Each particle i moves toward its personal best position $\mathbf{p}_i(h) = (p_{i1}, p_{i2}, \dots, p_{iD})$ ($pbest$), which defines their best previous position and the global best position $\mathbf{g}(h) = (g_1, g_2, \dots, g_D)$ ($gbest$), that defines the best $pbest$ with the best objective value among all particles, i.e., $\mathbf{g}(h) = \mathbf{p}_g(h)$, where g is a $gbest$ index defined by

$$g = \arg \min_i F(\mathbf{p}_i(h)), \quad (1)$$

where F is the objective function under consideration. Thus, $\mathbf{g}(h)$ is the global optimum solution at the iteration h . The elements of $\mathbf{v}_i$ and $\mathbf{z}_i$ in each dimension d of a particle i are updated as follows [12]:

$$\begin{aligned} v_{id}(h+1) &= wv_{id}(h) + r_1c_1(p_{id}(h) - z_{id}(h)) + r_2c_2(g_d(h) - z_{id}(h)), \\ z_{id}(h+1) &= z_{id}(h) + v_{id}(h+1), \end{aligned} \quad (2)$$

where $d = 1, 2, \dots, D$, and r_1 and r_2 are two random numbers uniformly distributed in the interval $[0, 1]$, taking a different value in each dimension d . The inertia weight w determines how much of the particle's previous velocity is preserved, and c_1 and c_2 are fixed positive acceleration coefficients. This study set these three parameters to their optimal default values [13], namely, $w = 0.729$ and $c_1 = c_2 = 1.494$. These processes are repeated until the user-defined stop criterion is reached. Thus, $\mathbf{g}(h_{\text{end}})$ is the solution obtained by PSO, where h_{end} is the iteration counts at which the stop criterion is satisfied.

Particles sometimes can be attracted to regions outside the feasible search space $\mathcal{S}$. One of the methods to preserve the feasibility of the found solution is in which particles located outside the

Algorithm 1 Pseudo code of the PSO.

▷ *Objective variable* $\mathbf{z}$
▷ *Objective function* $F(\mathbf{z})$.
Randomly initialize the position $\mathbf{z}$ of each particle i by uniformly distributing the particles over the search space $[z_{\min}, z_{\max}]^D$.
Initialize other particle variables as $\mathbf{v}_i(0) = 0$, $\mathbf{p}_i(0) = \mathbf{z}_i(0)$, $F(\mathbf{p}_i(0)) = \infty$ and $F(\mathbf{g}(0)) = \infty$.
Iteration counter $h \leftarrow 0$.
while ($h < H$ AND $F(\mathbf{g}(h)) > C$) **do**
 $h \leftarrow h + 1$
 for $i = 1$ to M , all M particles **do**
 Calculate $F(\mathbf{z}_i(h))$.
 if $F(\mathbf{z}_i(h)) < F(\mathbf{p}_i(h - 1))$ **then**
 Update p_{best} of particle i as $\mathbf{p}_i(h) = \mathbf{z}_i(h)$.
 else
 $\mathbf{p}_i(h) = \mathbf{p}_i(h - 1)$.
 end if
 end for i
 Update $\mathbf{g}_{best}$ as $\mathbf{g}(h) = \mathbf{p}_g(h)$, where $g = \arg \min_i F(\mathbf{p}_i(h))$.
 for $i = 1$ to M , all M particles **do**
 for $d = 1$ to D , all D dimensions **do**
 Update v_{id} and z_{id} according to Eq. (2).
 end for d
 end for i
end while
Output the final results $\mathbf{g}(h_{\text{end}})$.

search space cannot become its p_{best} even though its function value is better than that of previous p_{best} . Therefore, they are attracted back to the feasible search space in subsequent iterations.

Algorithm 1 is the pseudo code of the PSO that minimizes the optimization, where H is the maximum iteration and C is the stop criterion. Thus, PSO terminates the search process when $h > H$ or $F(\mathbf{g}(h)) < C$ is satisfied.

3. Definition of dynamical systems

3.1 Discrete-time dynamical system

A general form of the N -dimensional discrete-time dynamical system is described as

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \boldsymbol{\lambda}), \quad (3)$$

where k denotes discrete time, and $\mathbf{x}(k) \in \mathbb{R}^N$ and $\boldsymbol{\lambda} \in \mathbb{R}^L$ correspond to the state variables and system parameters, respectively.

Let $\mathbf{f}^l$ denote the l -th iteration of $\mathbf{f}$. A point $\mathbf{x}_p \in \mathbb{R}^N$ is said to be an n -periodic point of $\mathbf{f}$ if $\mathbf{x}_p = \mathbf{f}^n(\mathbf{x}_p, \boldsymbol{\lambda})$ and $\mathbf{x}_p \neq \mathbf{f}^l(\mathbf{x}_p, \boldsymbol{\lambda})$ for $l < n$. A 1-periodic point is referred to as a fixed point.

The Jacobian matrix of $\mathbf{f}$ at the n -periodic point $\mathbf{x}_p$ is described by

$$D\mathbf{f}^n(\mathbf{x}_p, \boldsymbol{\lambda}) = \prod_{l=0}^{n-1} \frac{\partial}{\partial \mathbf{x}(n-1-l)} (\mathbf{f}(\mathbf{x}(n-1-l), \boldsymbol{\lambda})), \quad (4)$$

with characteristic equation

$$\det(D\mathbf{f}^n(\mathbf{x}_p, \boldsymbol{\lambda}) - \mu \mathbf{I}_N) = 0, \quad (5)$$

where $\mathbf{x}(0) = \mathbf{x}_p$, $\mathbf{I}_N$ denotes the $N \times N$ identity matrix, and μ is a characteristic multiplier of $D\mathbf{f}^n(\mathbf{x}_p, \boldsymbol{\lambda})$. The characteristic multipliers determine the local stability class of an n -periodic point. If all characteristic multipliers of $D\mathbf{f}^n(\mathbf{x}_p, \boldsymbol{\lambda})$ are inside the unit circle in the complex plane, then $\mathbf{x}_p$

is a stable periodic point (SPP). If any characteristic multipliers of $D\mathbf{f}^n(\mathbf{x}_p, \boldsymbol{\lambda})$ are outside the unit circle, then $\mathbf{x}_p$ is an unstable periodic point (UPP). Period-doubling and saddle-node bifurcations occur under the conditions $\mu = -1$ and $\mu = 1$, respectively. Table II summarizes the variables used in the dynamical system definition.

3.2 Continuous dynamical system

We consider an N -dimensional continuous dynamical system described by ODEs in Eq. (6):

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(t, \mathbf{x}, \boldsymbol{\lambda}), \quad (6)$$

where $t \in \mathbb{R}$ denotes the time; $\mathbf{x} \in \mathbb{R}^N$ corresponds to the state variables and $\boldsymbol{\lambda} \in \mathbb{R}^L$ are the system parameters. Since the state velocity includes the time t , Eq. (6) is called a *non-autonomous system*.

The function $\mathbf{f} : \mathbb{R} \times \mathbb{R}^N \times \mathbb{R}^L \rightarrow \mathbb{R}^N$ is continuously differentiable, namely, smooth function (class C^∞). The map $\mathbf{f}$ periodically varies according to the time t with period T . A solution of an initial state $\mathbf{x}(t_0) = \mathbf{x}_0 \in \mathbb{R}^N$ ($t_0 = 0$) is defined as

$$\mathbf{x}(t) = \boldsymbol{\varphi}(t, \mathbf{x}_0, \boldsymbol{\lambda}). \quad (7)$$

Then the Poincaré map Φ is described as

$$\Phi : \mathbb{R}^N \rightarrow \mathbb{R}^N; \mathbf{x}_0 \mapsto \mathbf{x}_1 = \Phi(\mathbf{x}_0) = \boldsymbol{\varphi}(T, \mathbf{x}_0, \boldsymbol{\lambda}). \quad (8)$$

Let Φ^l denote the l -th iteration of the map Φ which is described as

$$\Phi^l : \mathbb{R}^N \rightarrow \mathbb{R}^N; \mathbf{x}_0 \mapsto \mathbf{x}_l = \Phi^l(\mathbf{x}_0) = \underbrace{\Phi \circ \Phi \circ \dots \circ \Phi}_{l \text{ times}}(\mathbf{x}_0) = \boldsymbol{\varphi}(lT, \mathbf{x}_0, \boldsymbol{\lambda}). \quad (9)$$

In other words, the Poincaré map converts the continuous system to the discrete-time system by sampling the orbit from the initial state $\mathbf{x}_0$ with the sampling interval T as

$$\{\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_l, \dots\} = \{\mathbf{x}_0, \Phi(\mathbf{x}_0), \Phi^2(\mathbf{x}_0), \dots, \Phi^l(\mathbf{x}_0), \dots\}. \quad (10)$$

An n -periodic solution of Φ is defined as follows:

$$\mathbf{x}_p = \Phi^n(\mathbf{x}_p) = \boldsymbol{\varphi}(nT, \mathbf{x}_p, \boldsymbol{\lambda}) \text{ and } \mathbf{x}_p \neq \Phi^l(\mathbf{x}_p) \text{ for } l < n. \quad (11)$$

The Jacobian matrix of Φ^n at the n -periodic solution $\mathbf{x}_p$ is described as

$$D\boldsymbol{\varphi}(nT, \mathbf{x}_p, \boldsymbol{\lambda}) = D\Phi^n(\mathbf{x}_p) = \frac{\partial \Phi^n(\mathbf{x}_p)}{\partial \mathbf{x}_p}, \quad (12)$$

with the characteristic equation:

$$\det(D\Phi^n(\mathbf{x}_p) - \mu \mathbf{I}_N) = 0. \quad (13)$$

By the way, in Eq. (12), $\partial \Phi^n(\mathbf{x}_p)/\partial \mathbf{x}_p$ is the matrix described by

$$\frac{\partial \Phi^n(\mathbf{x}_p)}{\partial \mathbf{x}_p} = \begin{pmatrix} \frac{\partial \varphi_1(nT, \mathbf{x}_p, \boldsymbol{\lambda})}{\partial x_{p1}} & \frac{\partial \varphi_1(nT, \mathbf{x}_p, \boldsymbol{\lambda})}{\partial x_{p2}} & \dots & \frac{\partial \varphi_1(nT, \mathbf{x}_p, \boldsymbol{\lambda})}{\partial x_{pN}} \\ \frac{\partial \varphi_2(nT, \mathbf{x}_p, \boldsymbol{\lambda})}{\partial x_{p1}} & \ddots & & \vdots \\ \vdots & & \ddots & \frac{\partial \varphi_{N-1}(nT, \mathbf{x}_p, \boldsymbol{\lambda})}{\partial x_{pN}} \\ \frac{\partial \varphi_N(nT, \mathbf{x}_p, \boldsymbol{\lambda})}{\partial x_{p1}} & \dots & \dots & \frac{\partial \varphi_N(nT, \mathbf{x}_p, \boldsymbol{\lambda})}{\partial x_{pN}} \end{pmatrix} \quad (14)$$

The remaining part of this section explains how to obtain the respective elements in Eq. (14) [4]. By substituting the solution starting from $\mathbf{x}_p$ into the system equation Eq. (6), we obtain

Table II. Variables and their symbols of the dynamical system.

Variable	Symbol
Time of the system (for discrete-time systems)	k
Time of the system (for continuous-time systems)	t
State variable vector	$\mathbf{x}$
Periodic point vector	$\mathbf{x}_p$
System parameter vector	$\boldsymbol{\lambda}$
Dimension of the system	N
Number of system parameters	L
Period on which the system depends	T
Iteration of the map	l
Number of periods of the periodic solution	n
Characteristic multiplier	μ

$$\frac{d\varphi(t, \mathbf{x}_p, \boldsymbol{\lambda})}{dt} = \mathbf{f}(t, \varphi(t, \mathbf{x}_p, \boldsymbol{\lambda}), \boldsymbol{\lambda}) \quad (15)$$

and

$$\varphi(0, \mathbf{x}_p, \boldsymbol{\lambda}) = \mathbf{x}_p. \quad (16)$$

Partially differentiate both sides of Eq. (15) with respect to $\mathbf{x}_p$:

$$\frac{\partial}{\partial \mathbf{x}_p} \left(\frac{d\varphi(t, \mathbf{x}_p, \boldsymbol{\lambda})}{dt} \right) = \frac{\partial}{\partial \mathbf{x}_p} (\mathbf{f}(t, \varphi(t, \mathbf{x}_p, \boldsymbol{\lambda}), \boldsymbol{\lambda})). \quad (17)$$

The order of differentiation on the left-hand side in Eq. (17) is commutative, and then we can obtain

$$\frac{d}{dt} \left(\frac{\partial \varphi(t, \mathbf{x}_p, \boldsymbol{\lambda})}{\partial \mathbf{x}_p} \right) = \frac{\partial \mathbf{f}(t, \varphi(t, \mathbf{x}_p, \boldsymbol{\lambda}), \boldsymbol{\lambda})}{\partial \mathbf{x}} \bigg|_{\mathbf{x}=\mathbf{x}_p} \frac{\partial \varphi(t, \mathbf{x}_p, \boldsymbol{\lambda})}{\partial \mathbf{x}_p}. \quad (18)$$

Similarly, from Eq. (16), the initial state

$$\frac{\partial \varphi(0, \mathbf{x}_p, \boldsymbol{\lambda})}{\partial \mathbf{x}_p} = \mathbf{I}_N. \quad (19)$$

Therefore, we can obtain all the elements in Eq. (14) by numerically solving Eq. (18) with initial value described in Eq. (19) from 0 to nT .

4. Bifurcation point detection strategy by using the nested layer PSO (NLPSO)

To detect bifurcation parameters at a periodic point, we must derive not only the system parameters satisfying the bifurcation conditions but also the periodic point depending on the bifurcation parameters. We proposed the NLPSO-based detection strategy [9–11] for detecting local bifurcation points. NLPSO is performed by two PSOs with nested structure to find the system parameters $\boldsymbol{\lambda}$ that produces a bifurcation phenomenon on an n -periodic solution $\mathbf{x}_p$ in a dynamical system $\mathbf{f}$. In the NLPSO algorithm, the main PSO (PSO_{bif}) seeks the bifurcation parameter $\boldsymbol{\lambda}$, and the inner PSO (PSO_{pp}) searches for the periodic point (or periodic solution) $\mathbf{x}_p$ depending on the system parameter information that is given by the position of the PSO_{bif} particles. Although NLPSO requires no carefully set initial system parameters or exact calculation, it can directly and accurately detect the bifurcation parameter without a Lyapunov exponent or a gradient-based method. Please note that our NLPSO-based detection method is for only non-degeneracy bifurcations.

The NLPSO algorithm is here described in detail. Table I summarizes the symbols used in NLPSO. Algorithm 2 is the pseudo code of the PSO_{bif}, which is our proposed NLPSO-based detection strategy for local bifurcation parameters. Algorithm 3 is the pseudo code of the PSO_{pp}, which is invoked

Algorithm 2 Pseudo code of the NLPSO for bifurcation parameter detection in a dynamical system.

▷ Objective variable $\mathbf{z}_b$ corresponding to bifurcation parameter λ on the system $\mathbf{f}$.
 ▷ Objective function $F_{\text{bif}}(\mathbf{z}_b)$.
 Iteration counter $h_1 \leftarrow 0$.
while ($h_1 < H_1$ AND $F_{\text{bif}}(\mathbf{g}_b(h_1)) > C_{\text{bif}}$) **do**
 $h_1 \leftarrow h_1 + 1$
 for $i = 1$ to M_{bif} , all M_{bif} particles **do**
 $\mathbf{x}_p^* \leftarrow \text{PSOPP}(\mathbf{z}_{b_i}(h_1))$
 ▷ Carry out the PSO_{pp} in Algorithm 3 with the argument $\mathbf{z}_{b_i}(h_1)$.
 Evaluate $F_{\text{bif}}(\mathbf{z}_{b_i}(h_1))$ with $\mathbf{x}_p^*$ which is a return value of the PSO_{pp} .
 if $F_{\text{bif}}(\mathbf{z}_{b_i}(h_1)) < F_{\text{bif}}(\mathbf{p}_{b_i}(h_1))$ AND $\mathbf{z}_{b_i} \in \mathcal{S}_b$ **then**
 Update p_{best} as $\mathbf{p}_{b_i}(h_1) = \mathbf{z}_{b_i}(h_1)$.
 else
 Keep p_{best} as $\mathbf{p}_{b_i}(h_1) = \mathbf{p}_{b_i}(h_1 - 1)$.
 end if
 end for i
 Update g_{best} $\mathbf{g}_b(h_1) = \mathbf{p}_{b_g}(h_1)$, where $g = \arg \min_i F_{\text{bif}}(\mathbf{p}_{b_i}(h_1))$.
 for $i = 1$ to M_{bif} , all M_{bif} particles **do**
 for $d = 1$ to L , all L dimensions **do**
 Update $v_{bid}(h_1)$ and $z_{bid}(h_1)$ according to Eq. (2).
 end for d
 end for i
end while
 Output the final results $\mathbf{g}_b(h_{1\text{end}})$ as a bifurcation parameter λ .

Algorithm 3 Pseudo code of PSO_{pp} that finds n -periodic point.

procedure PSOPP(λ)
 ▷ Objective variable $\mathbf{z}_p$ corresponding to n -periodic point $\mathbf{x}_p$.
 ▷ Objective function $F_{\text{pp}}(\mathbf{z}_p)$.
 Initialize particle information of the PSO_{pp} .
 Iteration counter $h_2 \leftarrow 0$.
while ($h_2 < H_2$ AND $F_{\text{pp}}(\mathbf{g}_p(h_2)) > C_{\text{pp}}$) **do**
 $h_2 \leftarrow h_2 + 1$
 for $i = 1$ to M_{pp} , all M_{pp} particles **do**
 Evaluate $F_{\text{pp}}(\mathbf{z}_{p_i}(h_2))$.
 Update p_{best} $\mathbf{p}_{p_i}(h_2)$ if $F_{\text{pp}}(\mathbf{z}_{p_i}(h_2)) < F_{\text{pp}}(\mathbf{p}_{p_i}(h_2))$.
 end for i
 Update g_{best} $\mathbf{g}_p(h_2) = \mathbf{p}_{p_g}(h_2)$, where $g = \arg \min_i F_{\text{pp}}(\mathbf{p}_{p_i}(h_2))$.
 for $i = 1$ to M_{pp} , all M_{pp} particles **do**
 for $d = 1$ to N , all N dimensions **do**
 Update $v_{pid}(h_2)$ and $z_{pid}(h_2)$ according to Eq. (2).
 end for d
 end for i
end while
return $\mathbf{g}_p(h_{2\text{end}})$ ▷ The deemed n -periodic point depending on λ is $\mathbf{g}_p(h_{2\text{end}})$.
end procedure

from the PSO_{bif} . We can obtain single bifurcation parameter set by using NLPSO according to the Algorithm 2. Figure 1 shows a flowchart representing a process for detecting multiple different bifurcation parameter sets. Firstly, as preprocessing, we define two objective functions F_{bif} and F_{pp} used in the NLPSO, and we are going to explain this later in detail. Secondly, we set the value of

Fig. 1. Flowchart to detect multiple bifurcation parameter sets.

μ as -1 or 1 , according to the type of target bifurcation that one would like to be found, and set the number of periods n of the target bifurcation. Next, we initialize PSO_{bif} information in the same way to the PSO. After initialization, we carry out the bifurcation parameter search according to the Algorithm 2. The output of the PSO_{bif} corresponds to single bifurcation parameter set, and we can obtain multiple bifurcation parameter sets by repeating the implementation of the NLPSO with different random initial states of the particle, according to Fig. 1. In this figure, run is the simulation counter and R_{max} is the maximum number of simulations, thus, the proposed method repeats the bifurcation point search until $R_{\text{max}} \leq run$ is satisfied.

The remaining part of this section explains two objective functions used in NLPSO in detail. Let us consider a detection of the bifurcation parameters in a discrete-time dynamical system $\mathbf{f}$ defined in Eq. (3). Since the PSO_{bif} seeks the bifurcation parameter $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_L)$, its particle position $\mathbf{z}_{b_i}$ serves the system parameter $\boldsymbol{\lambda}$. Thus, the PSO_{bif} search-space dimension D is equivalent to the number of the system parameters L to search, and the i -th particle position vector at the PSO_{bif} iteration count h_1 is denoted by $\mathbf{z}_{b_i}(h_1) = (z_{b_{i1}}, z_{b_{i2}}, \dots, z_{b_{iL}})$. Similarly, the $pbest$ and $gbest$ are denoted by $\mathbf{p}_{b_i}(h_1) = (p_{b_{i1}}, p_{b_{i2}}, \dots, p_{b_{iL}})$ and $\mathbf{g}_b(h_1) = (g_{b1}, g_{b2}, \dots, g_{bL})$. The objective function $F_{\text{bif}}(\mathbf{z}_b)$ is defined as

$$F_{\text{bif}}(\mathbf{z}_b) = \begin{cases} \left| \det \left(D\mathbf{f}^n(\mathbf{x}_p^*, \mathbf{z}_b) - \mu \mathbf{I}_N \right) \right|, & \text{if } F_{\text{pp}}(\mathbf{x}_p^*) < C_{\text{pp}}, \\ \infty, & \text{otherwise,} \end{cases} \quad (20)$$

where $\mathbf{x}_p^*$ is a deemed n -periodic point detected by the PSO_{pp}, F_{pp} is an objective function of the PSO_{pp} as will be described later, and C_{pp} is the criterion of the deemed periodic point $\mathbf{x}_p^*$. Equation (20) defines the bifurcation condition at the n -periodic point $\mathbf{x}_p$ according to Eq. (5), where $F_{bif}(\mathbf{z}_b) = 0$ means that $\mathbf{z}_b$ produces a bifurcation at $\mathbf{x}_p$. Thus, it is a minimization optimization problem with minimum at 0. The PSO_{bif} stops searching if the iteration count h_1 exceeds the maximum iteration H_1 or g_{best} of the PSO_{bif} satisfies $F_{bif}(\mathbf{g}_b) < C_{bif}$, where C_{bif} corresponds to the accuracy of the bifurcation point found. Thus, the solution $\mathbf{g}_b(h_{1end})$ is the bifurcation parameter detected by NLPSO, where h_{1end} is the iteration counts at which the PSO_{bif} satisfied the stop criterion. If we require only the bifurcation parameter located in the target parameter space, the PSO_{bif} specifies the feasible search space $\mathcal{S}_b$. Note that Eq. (20) guarantees an accuracy of the deemed periodic point $\mathbf{x}_p^*$ that composes of the detected bifurcation parameters λ . When a PSO_{bif} particle takes system parameters for which no n -periodic solution exists, it does not affect the p_{best} and g_{best} of the PSO_{bif} because its F_{bif} is assigned a bad value ($F_{bif} = \infty$). Therefore, the PSO_{bif} particles are attracted to other particles taking system parameters for which n -periodic solutions do exist. This effect avoids a false periodic point, that violates $F_{pp}(\mathbf{x}_p^*) < C_{pp}$, and is effective for the saddle-node bifurcation parameters detection [10].

To calculate Eq. (20), we must derive the n -periodic point $\mathbf{x}_p$ depending on the system parameter λ corresponding to the PSO_{bif} particle position $\mathbf{z}_b$. In the NLPSO, the λ -dependent periodic point is searched by the PSO_{pp} nested in the PSO_{bif}. Let M_{bif} denote the number of particles in the PSO_{bif}. By invoking the PSO_{pp} M_{bif} times, using different system parameters given by each $\mathbf{z}_{bi}$, the PSO_{pp} searches for the M_{bif} periodic solutions, which depend on the respective system parameters. The i -th particle position vector at the PSO_{pp} iteration count h_2 is denoted by $\mathbf{z}_{pi}(h_2) = (z_{pi1}, z_{pi2}, \dots, z_{piN})$, and it serves the λ -dependent periodic point $\mathbf{x}_p = (x_{p1}, x_{p2}, \dots, x_{pN})$. Thus, the PSO_{pp} search-space dimension D is equivalent to the dimension of the system N . Similarly, the p_{best} and g_{best} are denoted by $\mathbf{p}_{pi}(h_2) = (p_{pi1}, p_{pi2}, \dots, p_{piN})$ and $\mathbf{g}_p(h_2) = (g_{p1}, g_{p2}, \dots, g_{pN})$. The objective function of the PSO_{pp} is defined as

$$F_{pp}(\mathbf{z}_p) = \|\mathbf{f}^n(\mathbf{z}_p, \lambda) - \mathbf{z}_p\|, \quad (21)$$

where $\|\cdot\|$ denotes the Euclidean distance, and $\mathbf{f}^n(\mathbf{z}_p, \lambda)$ corresponds to the state variable, which defines the n -th iteration of $\mathbf{f}$ with initial point $\mathbf{z}_p$ and parameter λ (see Eq. (3)). Therefore, $F_{pp}(\mathbf{z}_p, \lambda)$ is minimized at 0 when $\mathbf{z}_p$ is exactly an n -periodic point (i.e. $\mathbf{z}_p \equiv \mathbf{x}_p$). The PSO_{pp} stops searching if the iteration count h_2 exceeds the maximum iteration H_2 or g_{best} of the PSO_{pp} satisfies $F_{pp}(\mathbf{g}_p) < C_{pp}$, where C_{pp} corresponds to the accuracy of the periodic point found. Thus, $\mathbf{g}_p(h_{2end})$ is the detected solution which is a return value of the PSO_{pp}, where h_{2end} is the iteration counts at which the PSO_{pp} satisfied the stop criterion. Meanwhile, the g_{best} $\mathbf{g}_p$ detected by the PSO_{pp} after searching, is called a *deemed n -periodic point* $\mathbf{x}_p^*$ because it is only an estimated periodic point and not guaranteed to be a true n -periodic point even if $\mathbf{x}_p^*$ satisfied the criterion $F_{pp}(\mathbf{x}_p^*) < C_{pp}$. In addition, if a PSO_{bif} particle takes system parameters λ with no n -periodic point, the PSO_{pp} can never detect a periodic point. In this case, the PSO_{pp} returns a false-periodic point, that violates $F_{pp}(\mathbf{x}_p^*) < C_{pp}$, to the PSO_{bif} [10].

Here, we make mention of a risk that PSO_{pp} finds a non-target periodic point that satisfies F_{pp} . If there is a periodic point with a non-target period k in the target parameter space, and k is a divisor of n , the non-target periodic point also satisfies Eq. (21). Therefore, on some bifurcation structure, as is the case with avoiding a false-periodic point it is better to give F_{bif} the ability to avoid non-target periodic points that are k -periodic points such that $k < n$.

Next, let us consider a bifurcation parameter detection in a non-autonomous system $\mathbf{f}$ defined in Eq. (6). Although the basic algorithm is the same as the case in the discrete-time dynamical systems, it can be extended to the non-autonomous system using the Poincaré map. The PSO_{bif} objective function F_{bif} to seek a bifurcation parameter in a non-autonomous system is defined as

$$F_{bif}(\mathbf{z}_b) = \begin{cases} |\det(D\Phi^n(\mathbf{x}_p^*) - \mu \mathbf{I}_N)|, & \text{if } F_{pp}(\mathbf{x}_p^*) < C_{pp}, \\ \infty, & \text{otherwise,} \end{cases} \quad (22)$$

where $\Phi^n(\mathbf{x}_p^*)$ is an n -periodic solution of a Poincaré map Φ ; thus $\Phi^n(\mathbf{x}_p^*) = \varphi(nT, \mathbf{x}_p^*, \mathbf{z}_b)$. Therefore, Eq. (22) is obtained by calculating the Poincaré map of the non-autonomous system.

The PSO_{pp} objective function F_{pp} to seek a λ -dependent periodic solution $\mathbf{x}_p$ in Eq. (22) is defined as

$$F_{pp}(\mathbf{z}_p) = \|\Phi^n(\mathbf{z}_p) - \mathbf{z}_p\|, \quad (23)$$

where $\Phi^n(\mathbf{z}_p) = \varphi(nT, \mathbf{z}_p, \lambda)$ defines the n -th iteration of Φ with initial point $\mathbf{z}_p$ and system parameter λ corresponding to a PSO_{bif} particle position $\mathbf{z}_b$. This is obtained by considering the Poincaré map in the same way as the PSO_{bif}.

5. Simulations

This section demonstrates some examples of 2-parameter bifurcation diagrams obtained by applying the NLPSO to a discrete-time dynamical system and a non-autonomous dynamical system.

5.1 Discrete-time dynamical system — Circle map

The circle map is a one-dimensional map described by

$$x(k+1) = f(x(k), \lambda) = \left(x(k) + \lambda_1 - \frac{\lambda_2}{2\pi} \sin 2\pi x(k) \right) \mod 1.0, \quad (24)$$

where $x(k)$ is a state variable, and $\lambda = (\lambda_1, \lambda_2)$ are system parameters. The circle map exhibits bifurcation phenomena as these parameters are varied.

From Eq. (20), the objective function of the PSO_{bif}, which detects the local bifurcation parameter set, is defined as

$$\begin{aligned} F_{bif}(\mathbf{z}_b) &= \begin{cases} \left| \frac{df(x_p^*, \mathbf{z}_b)}{dx_p^*} - \mu \right|, & \text{if } F_{pp}(x_p^*) < C_{pp} \\ \infty, & \text{otherwise} \end{cases} \\ &= \begin{cases} \left| \prod_{l=0}^{n-1} (1 - z_{b2} \cos 2\pi x(n-1-l)) - \mu \right|, & \text{if } F_{pp}(x_p^*) < C_{pp} \\ \infty, & \text{otherwise.} \end{cases} \end{aligned} \quad (25)$$

where $\mathbf{z}_b = (z_{b1}, z_{b2})$ serves $\lambda = (\lambda_1, \lambda_2)$ in Eq. (24); thus, $L = 2$. $x(k)$ is a state variable calculated by Eq. (24), and its initial value $x(0)$ is the deemed n -periodic point x_p^* depending on the parameter set (z_{b1}, z_{b2}) corresponding to the particle information of the PSO_{bif}.

The objective function of the PSO_{pp}, which detects $\mathbf{z}_b$ -dependent deemed periodic point x_p^* , is given by

$$F_{pp}(\mathbf{z}_p) = |f^n(z_{p1}, \lambda) - z_{p1}|, \quad (26)$$

where $z_p = z_{p1}$ serves $x(0)$ in Eq. (24); thus, $N = 1$. The system parameter vector λ in Eq. (26) is an argument of PSO_{pp} and corresponds to a position vector $\mathbf{z}_b$ of a single PSO_{bif} particle that invoked the PSO_{pp}.

Table III summarizes the experimental parameters of PSO_{bif} and PSO_{pp}. The search-space range of the PSO_{bif} is set as

Table III. Variables and parameters of the nested PSOs in NLPSO.

	PSO _{bif}	PSO _{pp}
Number of dimensions D	L	N
Stop criterion C_{bif}, C_{pp}	1×10^{-3}	1×10^{-5}
Maximum iterations H_1, H_2	300	100
Number of particles M_{bif}, M_{pp}		30
Inertia weight w		0.729
Acceleration coefficients c_1, c_2		1.494

Fig. 2. Parameter detection results of period-doubling bifurcations (I^n) and saddle-node bifurcations (G^n) in the circle map. $\mu = -1$ or 1 . $n = 2, 3$ or 5 .

Table IV. Simulation results of the circle map (averaged over 300 independent trials of NLPSO).

	n	PSO _{bif}				PSO _{pp}		
		Mean F_{bif}	SD F_{bif}	Mean $h_{1\text{end}}$	Suc[%]	Mean F_{pp}	SD F_{pp}	Suc[%]
I^n	2	5.12×10^{-4}	2.91×10^{-4}	24.51	100	5.02×10^{-6}	2.85×10^{-6}	100
	3	4.99×10^{-4}	2.82×10^{-4}	37.96	100	5.06×10^{-6}	2.74×10^{-6}	100
	5	4.52×10^{-4}	2.79×10^{-4}	53.13	100	4.94×10^{-6}	2.99×10^{-6}	100
G^n	2	5.02×10^{-4}	3.04×10^{-4}	78.95	100	6.18×10^{-6}	3.00×10^{-6}	100
	3	5.07×10^{-4}	2.72×10^{-4}	92.67	100	6.13×10^{-6}	3.05×10^{-6}	100
	5	4.91×10^{-4}	2.89×10^{-4}	79.59	100	6.45×10^{-6}	3.09×10^{-6}	100

$$\mathcal{S}_b : 0.4 \leq z_{b1} \leq 0.6 \wedge 0.5 \leq z_{b2} \leq 2.0, \quad (27)$$

and the initialization range of $\mathbf{z}_p$ is set as $[0, 1]$. For different number of periodic points ($n = 2$ -, 3 - and 5 -), NLPSO detected a period-doubling bifurcation parameter set ($\mu = -1$) denoted I^n and a saddle-node bifurcation set ($\mu = 1$) denoted G^n . These bifurcation parameters were obtained by the same algorithm with the same objective functions, requiring only a change of the parameters n and μ .

In Fig. 2, a two-dimensional bifurcation diagram of λ_1 versus λ_2 is plotted using the successfully detected bifurcation parameter sets $\boldsymbol{\lambda} = \mathbf{g}_b(h_{1\text{end}}) = (g_{b1}, g_{b2})$. Successful detection means that the detected bifurcation parameters and the periodic point composed of the bifurcation parameters satisfied both $F_{\text{bif}}(\mathbf{g}_b(h_{1\text{end}})) < C_{\text{bif}}$ and $F_{\text{pp}}(\mathbf{g}_p(h_{2\text{end}})) < C_{\text{pp}}$. This figure confirms the accurate detection of both the period-doubling bifurcation parameters and the saddle-node bifurcation parameters for all numbers of periods by the NLPSO-based detection method. Note that NLPSO detected the bifurcation parameters both stable periodic points and unstable periodic points without any careful initialization.

Table IV shows the results of $300 (= R_{\text{max}})$ simulations initialized in different random states for each bifurcation type and periodic number. The table shows the means and standard deviations (SDs) of $F_{\text{bif}}(\mathbf{g}_b(h_{1\text{end}}))$ and $F_{\text{pp}}(\mathbf{g}_p(h_{2\text{end}}))$ and the mean $h_{1\text{end}}$ of the PSO_{bif}. In this table, Suc[%] values of the PSO_{bif} and the PSO_{pp} denote the successful detection rate. The PSO_{bif} detected 100% of the period-doubling and saddle-node bifurcations, and the mean $h_{1\text{end}}$ was well below the iteration limit ($H_1 = 300$). Notably, the success rate of the PSO_{pp} was also 100%. These results mean that NLPSO successfully detected the local bifurcations with 100%. Moreover, the small SDs of F_{bif} and F_{pp} confirm the high robustness of NLPSO. Although NLPSO does not guarantee that a global optimum can be found, there is a high probability an acceptable solution will be found. Therefore,

Fig. 3. Optimization process of the PSO_{bif} which searched single 2-periodic saddle-node bifurcation parameter. Gray lines denote 2-periodic saddle-node bifurcation curves. (a) $F_{\text{bif}}(\mathbf{g}_b(h_1)) = 4.8457 \times 10^{-1}$. $(g_{b1}, g_{b2}) = (0.5313, 0.9854)$. (b) $F_{\text{bif}}(\mathbf{g}_b(h_1)) = 1.557 \times 10^{-1}$. $(g_{b1}, g_{b2}) = (0.5263, 0.8447)$. (c) $F_{\text{bif}}(\mathbf{g}_b(h_1)) = 5.5292 \times 10^{-2}$. $(g_{b1}, g_{b2}) = (0.4831, 0.6656)$. (d) $F_{\text{bif}}(\mathbf{g}_b(h_1)) = 3.812 \times 10^{-4} < C_{\text{bif}}$. $(g_{b1}, g_{b2}) = (0.4862, 0.5969)$.

even if NLPSO failed to search for a solution, NLPSO can find the accurate bifurcation point with high probability by re-searching with the different random initial state. These facts show that NLPSO requires no careful initialization as the Newton-based method.

In considering the NLPSO optimization process, Fig. 3 shows the movement of the PSO_{bif} particles that converged to the 2-periodic saddle-node bifurcation parameter $(\lambda_1, \lambda_2) = (0.4862, 0.5969)$. The yellow circles and red cross markers in the chart denote the position vectors $\mathbf{z}_b(h_1)$ of 30 particles in the PSO_{bif} and the $\mathbf{g}_{\text{best}} \mathbf{g}_b(h_1)$ at the iteration count h_1 , respectively. Since the particle position $\mathbf{z}_b$ of the PSO_{bif} corresponds to $\boldsymbol{\lambda}$ in the circle map as described in Eq. (24), the particles move on a system parameter space of the λ_1 - λ_2 plane. Gray lines denote the 2-periodic saddle-node bifurcation curves [1]. In other words, the multiple optimum solutions, in which $F_{\text{bif}}(\boldsymbol{\lambda}) = 0$, co-exist on the gray lines. On the other hand, Fig. 4 shows a search space of the PSO_{pp} for the $\text{PSO}_{\text{bif}} \mathbf{g}_{\text{best}}$ at the iteration count h_1 , and this figure is a return map of the circle map onto $x(k)$ - $x(k+2)$ plane for the system parameter $\boldsymbol{\lambda}$ that is the $\text{PSO}_{\text{bif}} \mathbf{g}_{\text{best}} \mathbf{g}_b(h_1)$ at h_1 . Black markers denote the 2-periodic orbit $x_p^* = g_p$ detected by the PSO_{pp} . Since the particle position $\mathbf{z}_p$ of the PSO_{pp} corresponds to

Fig. 4. Return map of the circle map onto the $x(k)$ - $x(k+2)$ plane for the parameter set λ that is the g_{best} $\mathbf{g}_b(h_1)$ of the PSO_{bif} at the iteration count h_1 . Black circle denotes the 2-periodic point $x_p^* = g_p$ detected by the PSO_{pp} . (a) $(\lambda_1, \lambda_2) = (0.5313, 0.9854)$. $g_p = 0.11585$. (b) $(\lambda_1, \lambda_2) = (0.5263, 0.8447)$. $g_p = 0.18687$. (c) $(\lambda_1, \lambda_2) = (0.4831, 0.6656)$. $g_p = 0.40586$. (d) $(\lambda_1, \lambda_2) = (0.4862, 0.5969)$. $g_p = 0.40970$.

$x(0)$ in Eq. (24), the particles move on a line of the state variable x . In the initial state of the PSO_{bif} at $h_1 = 0$ (Fig. 3(a)), the PSO_{bif} particles were randomly spread in the target parameter space. In this way, we do not need to carefully set the initial value to search, but we only need to set it randomly. As the iterations proceeded, the particles swarmed toward one of the G^2 bifurcation parameters as shown in Figs. 3(b)–(c). Meanwhile, Figs. 4(a)–(c) show that a shape of optimization space of F_{pp} varies with the $\mathbf{g}_b(h_1)$ value. This shows that the PSO_{pp} should be used for each PSO_{bif} particle since the PSO_{bif} particles take a different periodic point x_p^* depending on respective particle position $\mathbf{z}_{bi}$. Furthermore, the PSO_{pp} found both SPPs and UPPs without careful initialization. If multiple SPPs and/or UPPs coexist in the map, the PSO_{pp} particles converge to one of the periodic points, depending on the PSO_{pp} initial state and the random number values in the updated equation (Eq. (2)). In the searching process, if a PSO_{bif} particle takes parameters for which no 2-periodic point exists, the PSO_{pp} returns a *false* periodic point that lies outside the criterion C_{pp} for that particle. Then, the PSO_{bif} assigns a bad F_{bif} value (∞) to that particle. Since PSO steers the poor-scoring particles toward those with good F_{bif} values, the PSO_{bif} particles automatically move to the parameter area containing the correct 2-periodic points, without requiring special manipulation or careful initialization. Furthermore, since PSO_{bif} particles located outside the target parameter space cannot become its p_{best} , they are attracted back to the target parameter space in subsequent iterations. At $h_1 = 61$ (Fig. 3(d)), the PSO_{bif} that found a saddle-node bifurcation point satisfied

the $F_{\text{bif}} < 10^{-3}$ criterion, which terminated the bifurcation point searching. Obviously, the PSO_{bif} particles located outside the target parameter space never become the *gbest*. Therefore, the detected bifurcation parameter certainly locates inside the target parameter space.

5.2 Non-autonomous system — Duffing equation

Next, we apply NLPSO to the Duffing equation, which is a non-autonomous system and exhibits chaotic behavior. The system is given by

$$\frac{dx_1}{dt} = x_2 = f_1(t, \mathbf{x}, \boldsymbol{\lambda}), \quad (28)$$

$$\frac{dx_2}{dt} = -0.2x_2 - x_1^3 + \lambda_2 + \lambda_1 \cos(t) = f_2(t, \mathbf{x}, \boldsymbol{\lambda}). \quad (29)$$

where $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ is a vector of state variables, and $\boldsymbol{\lambda} = (\lambda_1, \lambda_2) \in \mathbb{R}^2$ is a vector of system parameters. Equation (29) periodically varies according to the time t with period $T = 2\pi$, so the Poincaré section is at $t = 2\pi$.

The objective function of the PSO_{bif} is described by

$$F_{\text{bif}}(\mathbf{z}_b) = \begin{cases} \left| \det \left(\frac{\partial \Phi^n}{\partial \mathbf{x}_p^*} (\mathbf{x}_p^*) - \mu \mathbf{I}_N \right) \right|, & \text{if } F_{\text{pp}}(\mathbf{x}_p^*) < C_{\text{pp}} \\ \infty, & \text{otherwise,} \end{cases}$$

$$= \begin{cases} \left| \det \left(\begin{bmatrix} \frac{\partial \varphi_1}{\partial x_{p1}^*} (2n\pi, \mathbf{x}_p^*, \mathbf{z}_b) & \frac{\partial \varphi_1}{\partial x_{p2}^*} (2n\pi, \mathbf{x}_p^*, \mathbf{z}_b) \\ \frac{\partial \varphi_2}{\partial x_{p1}^*} (2n\pi, \mathbf{x}_p^*, \mathbf{z}_b) & \frac{\partial \varphi_2}{\partial x_{p2}^*} (2n\pi, \mathbf{x}_p^*, \mathbf{z}_b) \end{bmatrix} - \mu \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \right|, & \text{if } F_{\text{pp}}(\mathbf{x}_p^*) < C_{\text{pp}}, \\ \infty, & \text{otherwise,} \end{cases} \quad (30)$$

where $\mathbf{z}_b = (z_{b1}, z_{b2})$ is the particle position vector of the PSO_{bif} and serves $\boldsymbol{\lambda}$ in Eq. (29); thus, $L = 2$. The initial value $\mathbf{x}_0 (= \mathbf{x}(0))$ is a deemed n -periodic point $\mathbf{x}_p^* = (x_{p1}^*, x_{p2}^*)$ depending on the parameter set (z_{b1}, z_{b2}) and it is detected by the PSO_{pp} with $\mathbf{z}_b$. The system equation and the state variables $\mathbf{x}$ can be described as

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(t, \mathbf{x}, \mathbf{z}_b), \quad (31)$$

$$\mathbf{x}(t) = \boldsymbol{\varphi}(t, \mathbf{x}_p^*, \mathbf{z}_b). \quad (32)$$

Thus,

$$\boldsymbol{\varphi}(0, \mathbf{x}_p^*, \mathbf{z}_b) = \mathbf{x}_p^*. \quad (33)$$

The elements of Jacobian matrix in Eq. (30) can be obtained by solving following ODEs with the system equations in Eq. (29):

$$\frac{d}{dt} \begin{bmatrix} \frac{\partial \varphi_1}{\partial x_{p1}^*} \\ \frac{\partial \varphi_2}{\partial x_{p1}^*} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} \bigg|_{\mathbf{x}=\mathbf{x}_p^*} \begin{bmatrix} \frac{\partial \varphi_1}{\partial x_{p1}^*} \\ \frac{\partial \varphi_2}{\partial x_{p1}^*} \end{bmatrix} = \begin{bmatrix} \frac{\partial \varphi_2}{\partial x_{p1}^*} \\ -3x_{p1}^{*2} \frac{\partial \varphi_1}{\partial x_{p1}^*} - 0.2 \frac{\partial \varphi_2}{\partial x_{p1}^*} \end{bmatrix},$$

$$\frac{d}{dt} \begin{bmatrix} \frac{\partial \varphi_1}{\partial x_{p2}^*} \\ \frac{\partial \varphi_2}{\partial x_{p2}^*} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} \bigg|_{\mathbf{x}=\mathbf{x}_p^*} \begin{bmatrix} \frac{\partial \varphi_1}{\partial x_{p2}^*} \\ \frac{\partial \varphi_2}{\partial x_{p2}^*} \end{bmatrix} = \begin{bmatrix} \frac{\partial \varphi_2}{\partial x_{p2}^*} \\ -3x_{p1}^{*2} \frac{\partial \varphi_1}{\partial x_{p2}^*} - 0.2 \frac{\partial \varphi_2}{\partial x_{p2}^*} \end{bmatrix}, \quad (34)$$

$$\left[\begin{array}{c} \frac{\partial \varphi_1}{\partial x_{p1}^*} \\ \frac{\partial \varphi_2}{\partial x_{p1}^*} \end{array} \right] \bigg|_{t=0} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \left[\begin{array}{c} \frac{\partial \varphi_1}{\partial x_{p2}^*} \\ \frac{\partial \varphi_2}{\partial x_{p2}^*} \end{array} \right] \bigg|_{t=0} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Similarly, the objective function of the PSO_{pp}, which detects (λ_1, λ_2) -dependent n -periodic solutions, is given by

$$\begin{aligned} F_{pp}(\mathbf{z}_p) &= \|\Phi^n(\mathbf{z}_p) - \mathbf{z}_p\|, \\ &= \sqrt{(\varphi_1(2n\pi, \mathbf{z}_p, \boldsymbol{\lambda}) - z_{p1})^2 + (\varphi_2(2n\pi, \mathbf{z}_p, \boldsymbol{\lambda}) - z_{p2})^2}, \end{aligned} \quad (35)$$

where $\mathbf{z}_p = (z_{p1}, z_{p2})$ is the PSO_{pp} particle position vector and serves $\mathbf{x}_0$ in Eq. (29); thus, $N = 2$. The system parameter vector $\boldsymbol{\lambda}$ in Eq. (35) corresponds to a position vector $\mathbf{z}_b$ of a single PSO_{bif} particle that invoked the PSO_{pp}.

The experimental parameters of PSO_{bif} and PSO_{pp} are same as in Table III. For different number of periodic solutions ($n = 1$ - and 2 -), the NLPSO detected a period-doubling bifurcation parameter ($\mu = -1$) and a saddle-node bifurcation parameter ($\mu = 1$). The search-space range of the PSO_{bif} is set as

$$\mathcal{S}_b : 0.4 \leq z_{b1} \leq 0.6 \wedge 0.5 \leq z_{b2} \leq 2.0, \text{ for } I^n \quad (36)$$

$$\mathcal{S}_b : 0.2 \leq z_{b1} \leq 0.5 \wedge 0.5 \leq z_{b2} \leq 2.0, \text{ for } G^n \quad (37)$$

The initialization range of $\mathbf{z}_p$ is set as $[-0.4, 0.4]^N$. We computed these bifurcation parameters in 300 simulations with different random initial states of the particles.

Figure 5 is a two-dimensional bifurcation diagram of the respectively detected 300 bifurcation parameter sets [11]. This figure confirms that the bifurcation parameter sets are obtained correctly on both the period-doubling and saddle-node bifurcation parameters for all numbers of periods because the detected parameters are located on the bifurcation curves detected by Kawakami method [1] as shown in Fig. 6. Table V summarizes the results of 300 simulations of each bifurcation type and periodic number. The success rate of both PSO_{bif} and PSO_{pp} were 100%, and the mean $h_{1\text{end}}$ was below the iteration limit ($H_1 = 300$). The small SDs of F_{bif} denotes the robustness of NLPSO. We

Fig. 5. Parameter detection results of period-doubling bifurcations (I^n) and saddle-node bifurcations (G^n) in the Duffing equation. Three hundred points each. $\mu = -1$ or 1 . $n = 1$ or 2 .

Fig. 6. Two-dimensional bifurcation diagram of the Duffing equation, plotted by the Kawakami method [1].

Table V. Simulation results of the Duffing equation (averaged over 300 independent trials of NLPSO).

	n	PSO _{bif}				PSO _{pp}		
		Mean F_{bif}	SD F_{bif}	Mean $h_{1\text{end}}$	Suc[%]	Mean F_{pp}	SD F_{pp}	Suc[%]
I^n	1	4.71×10^{-4}	2.86×10^{-4}	23.0	100	6.66×10^{-6}	2.37×10^{-6}	100
	2	4.87×10^{-4}	2.84×10^{-4}	53.47	100	6.91×10^{-6}	2.18×10^{-6}	100
G^n	1	5.14×10^{-4}	2.98×10^{-4}	86.77	100	7.14×10^{-6}	2.38×10^{-6}	100

should mention that the obtained bifurcation points shown in Fig. 5 are ununiformly distributed. This is caused by a PSO characteristic that the particles are attracted to a search area where solutions that can be easier found exist. By limiting the search-space range of the PSO_{bif} to the area we want to obtain and combining the bifurcation parameters detected with different search-space range, the NLPSO can obtain uniformly distributed bifurcation parameters.

6. Conclusions

This paper explained in detail the NLPSO-based detection method of the 2-parameter bifurcation diagram in a discrete-time dynamical system and a non-autonomous system. Since this method does not use the derivative of the objective function, it requires no careful initialization or Lyapunov exponents. It is particularly worth noting that the NLPSO-based method does not require second-order partial derivative in non-autonomous systems. It means the high versatile usability of the NLPSO. In contrast to the traditional Newton method-based approaches, our NLPSO-based method can detect bifurcation parameters of nondifferentiable/discontinuous systems since PSO can realize global search and does not require differentiability of objective functions. These features suggest that the NLPSO-based method will be developed into an efficient tool for detecting high-dimensional periodic points in high-dimensional systems with large parameter spaces. Furthermore, NLPSO can detect the bifurcation parameter sets at which stable and unstable periodic point merges. The NLPSO can make a 2-parameter bifurcation diagram by itself, however, by assigning the bifurcation parameter, detected by the NLPSO, as the initial point of a conventional curve tracing, we could solve the initial value problem of the conventional bifurcation analysis.

We hope to extend the NLPSO into the bifurcation analysis in various dynamical systems, circuit design, chaos control, and other applications. This study has used the standard PSO since this type

of PSO is the simplest and includes only a fraction of parameters. However, we should compare the standard PSO used in this study to modified PSOs including deterministic PSO [14–16] or other metaheuristic optimizers [17], such as the Nelder-Mead simplex method and differential evolution [18]. Moreover, we also would like to develop a NLPSO-based bifurcation curve tracing method and a MATLAB/Python toolbox or software to carry out our method. It would be a great help to researchers specializing in the study of the dynamical systems.

Acknowledgments

We also thank Dr. Y. Miino for lending his expertise on making the bifurcation diagram by the Kawakami method. This work was supported by JSPS KAKENHI Grant Numbers 19K12138, 19K12106 and 19K12144. This research was supported by a grant from the Hibi Science Foundation.

References

- [1] H. Kawakami, “Bifurcation of periodic responses in forced dynamic nonlinear circuits: Computation of bifurcation values of the system parameters,” *IEEE Trans. on Circuits and Systems*, vol. 31, no. 3, pp. 248–260, 1984.
- [2] T. Ueta, M. Tsueike, H. Kawakami, T. Yoshinaga, and Y. Katsura, “A computation of bifurcation parameter values for limit cycles,” *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E80-A, no. 9, pp. 1725–1728, 1997.
- [3] T. Kousaka, T. Ueta, and H. Kawakami, “Bifurcation of switched nonlinear dynamical systems,” *IEEE Trans. on Circuits and Systems II*, vol. 2, no. 7, pp. 1451–1458, 1999.
- [4] K. Tsumoto, T. Ueta, T. Yoshinaga, and H. Kawakami, “Bifurcation analyses of nonlinear dynamical systems: From theory to numerical computations,” *NOLTA*, vol. 3, no. 4, pp. 458–476, 2012.
- [5] E.J. Doedel and B.E. Oldeman, 2012. <http://indy.cs.concordia.ca/auto/>.
- [6] F. Dercole and Y.A. Kuznetsov, “SlideCont: An auto97 driver for bifurcation analysis of Filippov systems,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 31, no. 1, pp. 95–119, 2005.
- [7] ERATO, JST, 2004. <http://bunki.ait.tokushima-u.ac.jp:50080/>.
- [8] J. Kennedy and R. Eberhart, “Particle swarm optimization,” *Proc. of IEEE Int. Joint Conf. on Neural Netw.*, pp. 1942–1948, 1995.
- [9] H. Matsushita, Y. Tomimura, H. Kurokawa, and T. Kousaka, “Period Doubling Bifurcation Point Detection Strategy with Nested Layer Particle Swarm Optimization,” *International Journal of Bifurcation and Chaos*, vol. 27, no. 7, 1750101, 2017.
- [10] H. Matsushita, H. Kurokawa, and T. Kousaka, “Saddle-node bifurcation parameter detection strategy with nested-layer particle swarm optimization,” *Chaos, Solitons & Fractals*, vol. 119, pp. 126–134, 2019.
- [11] H. Matsushita, H. Kurokawa, and T. Kousaka, “Nested-layer particle swarm optimization method for bifurcation point detection in non-autonomous systems,” *NOLTA*, vol. 10, no. 3, pp. 289–302, 2019.
- [12] Y. Shi and R. Eberhart, “A Modified Particle Swarm Optimizers,” *Proc. of IEEE Int. Conf. on Evolutionary Computation*, pp. 69–73, 1998.
- [13] I.C. Trelea, “The particle swarm optimization algorithm: convergence analysis and parameter selection,” *Information Processing Letters*, vol. 85, no. 6, pp. 317–325, 2003.
- [14] C.W. Cleghorn and A.P. Engelbrecht, “A generalized theoretical deterministic particle swarm model,” *Swarm Intelligence*, vol. 8, no. 1, pp. 35–59, 2014.
- [15] Y. Yamanaka and T. Tsubone, “An optimizer using swarm of chaotic dynamical particles,” *NOLTA*, vol. 6, no. 1, pp. 112–130, 2015.
- [16] T. Sasaki, H. Nakano, A. Miyauchi, and A. Taguchi, “Deterministic particle swarm optimizer with the convergence and divergence dynamics,” *IEICE Transactions on Fundamentals of Elec-*

- tronics, Communications and Computer Sciences*, vol. 100, no. 5, pp. 1244–1247, 2017.
- [17] I. Boussaïd, J. Lepagnot, and P. Siarry, “A survey on optimization metaheuristics,” *Information Sciences*, vol. 237, pp. 82–117, 2013.
 - [18] R. Storn and K. Price, “Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.