

Article

A Chaotic Krill Herd Optimization Algorithm for Global Numerical Estimation of the Attraction Domain for Nonlinear Systems

Messaoud Aloui ¹, Faïçal Hamidi ¹, Housseem Jerbi ^{2,*} , Mohamed Omri ³, Dumitru Popescu ⁴
and Rabeh Abbassi ⁵ 

¹ Laboratory “Modélisation, Analyse et Commande des Systèmes”, University of Gabes, Gabes LR16ES22, Tunisia; aloui_messaoud@yahoo.com (M.A.); faical.hamidi@isimg.tn (F.H.)

² Department of Industrial Engineering, College of Engineering, University of Hail, Hail 1234, Saudi Arabia

³ Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah 21589, Saudi Arabia; Mnomri@kau.edu.sa

⁴ Faculty of Automatics and Computers, University Politehnica of Bucharest, RO-060042 Bucharest, Romania; dumitru.popescu@acse.pub.ro

⁵ Department of Electrical Engineering, College of Engineering, University of Hail, Hail 1234, Saudi Arabia; r.abbassi@uoh.edu.sa

* Correspondence: h.jerbi@uoh.edu.sa

Abstract: Nowadays, solving constrained engineering problems related to optimization approaches is an attractive research topic. The chaotic krill herd approach is considered as one of most advanced optimization techniques. An advanced hybrid technique is exploited in this paper to solve the challenging problem of estimating the largest domain of attraction for nonlinear systems. Indeed, an intelligent methodology for the estimation of the largest stable equilibrium domain of attraction established on quadratic Lyapunov functions is developed. The designed technique aims at computing and characterizing a largest level set of a Lyapunov function that is included in a particular region, satisfying some hard and delicate algebraic constraints. The formulated optimization problem searches to solve a tangency constraint between the LF derivative sign and constraints on the level sets. Such formulation avoids possible dummy solutions for the nonlinear optimization solver. The analytical development of the solution exploits the Chebyshev chaotic map function that ensures high search space capabilities. The accuracy and efficiency of the chaotic krill herd technique has been evaluated by benchmark models of nonlinear systems. The optimization solution shows that the chaotic krill herd approach is effective in determining the largest estimate of the attraction domain. Moreover, since global optimality is needed for proper estimation, a bound type meta-heuristic optimization solver is implemented. In contrast to existing strategies, the synthesized technique can be exploited for both rational and polynomial Lyapunov functions. Moreover, it permits the exploitation of a chaotic operative optimization algorithm which guarantees converging to an expanded domain of attraction in an essentially restricted running time. The synthesized methodology is discussed, with several examples to illustrate the advantageous aspects of the designed approach.

Keywords: nonlinear system; quadratic Lyapunov function; optimization; chaotic krill herd algorithm



Citation: Aloui, M.; Hamidi, F.; Jerbi, H.; Omri, M.; Popescu, D.; Abbassi, R. A Chaotic Krill Herd Optimization Algorithm for Global Numerical Estimation of the Attraction Domain for Nonlinear Systems. *Mathematics* **2021**, *9*, 1743. <https://doi.org/10.3390/math9151743>

Academic Editor: Cristina I. Muresan

Received: 29 May 2021

Accepted: 19 July 2021

Published: 23 July 2021

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The main problem of theory of optimization consists in selecting an optimal vector within a given space of search. The attained vector can minimize or maximize an objective function which is expected to offer the optimal solution. Mostly, advanced intelligent techniques are developed to deal with the design of these kinds of optimization processes. In view of their methodology, optimization techniques can be classified into several main categories: (1) analytical or deterministic, (2) heuristic or random advanced methods and (3) Multi-Objective or single objective optimization [1]. The first class, based mainly on

the gradient theory, are known to be “strict step” methods. Indeed, identical solutions are obtained whenever the initial starting conditions of the algorithm are the same. However, heuristic algorithms are founded on a random walks principle. As a result, the optimization approaches cannot be reiterated under any initializing conditions. Nonetheless, both classes mostly provide satisfactory solutions leading to similar final optimal results. Lately, bio-inspired algorithms demonstrate an impressive and valuable performance in solving challenging nonlinear optimization problems [2].

All meta-heuristic methods, to some degree, manage to identify a balance between local and global searches (intensification and randomization) [3]. To solve high-dimension nonlinear optimization problems including task-resource assignment, these flexible nature-inspired meta-heuristic methods are used. These approaches can fully use the entire data pertaining to the population in order to identify precise solutions. Evolution theory has received a lot of attention to date. Genetic algorithms (GAs) [4,5] are based on the gradient-free approach that mimics evolution. Since then, several meta-heuristic nature-inspired techniques have been formulated; these include particle swarm optimization (PSO) [6–8], evolutionary strategy (ES) [9], firefly algorithm (FA) [10], ant colony optimization (ACO) [11], differential evolution (DE) [12], probability-based incremental learning (PBIL) [13], big bang–big crunch algorithm [14], biogeography-based optimization (BBO) [15], harmony search (HS) [16], cuckoo search (CS) [17], animal migration optimization (AMO) [18], krill herd method (KH) [19], bat algorithm (BA) [20], teaching–learning-based optimization (TLBO) [21], and charged system search (CSS) [22]. KH is a modern swarm intelligence optimization technique inspired by krill herding behaviour [23]. In KH, the krill movement objective function is determined by the krill distance, krill swarm density, and food density. Krill location comprises three mechanisms: (i) movement due to other krill, (ii) foraging movement, and (iii) physical diffusion. The KH algorithm is beneficial because of its simplicity, facilitating easy implementation in parallel computing [24].

Additionally, KH is a new population-dependent swarm intelligence algorithm [24] considering the Lagrangian and evolutionary activities of krill in their natural environment. This algorithm can help understand and use krill behaviour for optimizing real-world problems. Although the KH based algorithm is convenient for many optimization problems, it cannot avoid local optima and therefore is inappropriate to determine a global optimal solution [25]. Several techniques are used in the literature to enhance the fundamental KH technique and improve its performance [26].

In order to address this concern, Wang et al. [27] suggested using an enhanced algorithm based on chaotic patterns. Chaotic maps have been implemented for preventing local optima and directly reaching the globally optimal solution [24]. Furthermore, this technique requires less computation time to reach a solution. Hence, it is feasible to avoid local optima and provide rapid convergence, which makes this technique usable in real-world applications for addressing optimization problems with distinct conditions.

This study introduces the Chaotic KH-based technique (CKH) for increasing the KH convergence rate. Numerous one-dimensional chaotic maps replace KH parameters. To generate chaotic sequences efficiently and rapidly, Chebyshev maps has been implemented in the CKH algorithm. The suggested technique is tested on benchmark problems in order to estimate the domain of attraction (DA). It is expected that CKH will provide better performance than the techniques like those developed in [28,29], specified in the literature. The primary reason for performance increase is the use of deterministic chaotic patterns instead of linearly declining values.

Stability is the most critical factor in the study of nonlinear dynamic systems. For most applications, knowing the asymptotic stability associated with an equilibrium position is typically inadequate because it is essential to consider the maximum distance between the initial state and equilibrium position to ensure asymptotic stability. Consequently, the principle of the domain of attraction was born. Since the publication of the stability theorem in 1892, the challenge of precisely evaluating or estimating the DA has remained unresolved. In general, evaluating the DA is very complex [30,31]; hence, the problem

concerning the computation of the most significant invariant DA subset is also taken into account. Lyapunov and non-Lyapunov methods are the two types of DA estimation methods [30].

The maximal Lyapunov function (LF) can be used to calculate the precise DA of a nonlinear system. A rational LF can be used to approximate such an expression [31,32]. Lyapunov quadratic functions typically compute stability area approximations conservatively. Regardless, quadratic LFs are widely employed for stabilization [33]. Moreover, these functions are used to build asymmetric LF used for forecasting DA associated with linear systems concerning asymmetrical extrinsic action [34].

In contrast, research indicates that random LFs fulfilling Lyapunov constraints offer DA estimates concerning control systems of the nonlinear polynomial type [35]. The set of LFs appropriate for computing DA was augmented using a proposition proved in [36]. A novel technique for computing LFs was formulated; its degrees of freedom were higher than the traditional technique [37]. Imprecise polynomial expressions concerning non-polynomial systems must be estimated in order to apply Lyapunov polynomial functions for computing the DA specific to non-polynomial systems [38]. Several approximation techniques that rely on the multi-parameter polynomial method to assess residual expression were formulated in [39]. The DA can also be computed using piecewise affine functions used as LFs [40–42].

Presently, nonlinear system stability and formulation of new DA estimates typically rely on the Pupov criterion, The Zubov technique [43], LaSalle's theorem, the circular criterion, and the framework of central manifolds [29].

In a majority of the situations, the problem concerning DA estimation is simplified to a convex or non-convex optimization problem [44]. This is addressed using optimization techniques such as SOS [45], LMI [46], intelligent optimization techniques [47], integration of the genetic algorithm and LMI [4]. In comparison to optimization approaches, a swift and computationally efficient technique formulated for real-time applications relies on an algorithm that identifies the most-extensive sublevel set pertaining to the LF under the constraint its time derivative is negative [28].

Predicting DA pertaining to inexact scenarios is highly challenging, specifically for situations comprising large scale systems [48]. An internal DA approximation pertaining to a type of inexact nonlinear systems can be determined using a novel technique that LFs independently of parameters [49]; moreover, the branch and bound technique is also used [50]. Furthermore, researchers [51] proposed an updated technique to compute sliding mode control stability under uncertain conditions. Filling measures have been widely used for stochastic control applications; these measures are required for computing the DA [52].

DA approximation is also performed using the non-Lyapunov trajectory inversion technique that relies on topological aspects. However, computational complexity is one challenge associated with this technique; this can be addressed by integrating the approach with Lyapunov techniques [53,54]. Formulating reachable sets is a different approach sometimes used for evaluation of DA. Formulating reachable sets using an approximate calculation method is specified in [55].

This paper suggests a metaheuristic scheme for Lyapunov-based methods to determine DA corresponding to several nonlinear systems. This technique is intended to have better computational effectiveness and to be helpful for real-time implementations. When a potential LF is selected, an evolutionary algorithm explores the biggest sublevel LF set under the constraint that the derivative with respect to time is negative for all values in the sublevel set. A heuristic optimization scheme is set up, comprising a tangency restriction concerning level set and LF sign requirements. These restrictions facilitate skipping numerous potential local solutions specific to the nonlinear optimization process.

In [56] a hybridized monarch butterfly optimization method has been proposed. Two state-of-the-art swarm intelligence approaches have been implemented and the designed technique was enhanced to be applied to a convolutional neural network scheme problem. In [57], the authors created a general model, the adaptive network-based fuzzy inference

system code, that has been applied for estimating nanofluids' relative viscosity. A statistical analysis has proved the precision of the synthesized model.

The main contribution of this paper consists in combining the CKH algorithm with position updating equations in a chaotic map to compute the best solution related to the DA estimation problem. The implemented CKH algorithm in this work will guarantee a high convergence speed in computing the radius of the maximal estimated DA. Moreover, the coordinates of the tangency point between the LF and its derivative will be computed and efficiently determined. Along with this valuable performance, a minimal optimization routine time budget will be insured, favoring trajectory tracking engineering control problems. To obtain the maximal expanded DA, a particular optimization problematic is investigated in this work. This later includes a delicate tangency constraint related to the LF sign and its derivative. The challenging objective consists in avoiding important potential dummy and local solutions for the applied optimization routine. The designed strategy will permit the tangency point to be fixed accurately and converge to the solution in a reduced running time. The assumption behind the investigated research is that the designed method could tackle problems better at hand than peer methods; indeed, no efficient procedure has been formulated to determine this particular point.

This paper discusses a novel approach for modifying the shape and increasing the size of the DA specific to nonlinear systems. Section 2 details how iterative techniques are set up to formulate the LF that optimizes DA. Section 3 details the CKH-specific parameter choice. Section 4 details simulation results of prevalent numerical problems. Section 5 presents a comparison of the proposed algorithm with other techniques; additionally, there are analyses of the results and discussion concerning algorithmic efficiency assessment. Summarizing conclusion and perspectives are provided in Section 6. An appendix comprising an overall discussion concerning the class of systems and several basic notions has been provided at the end of the paper.

2. Estimation of the Domain of Attraction

2.1. Fundamentals of the Domain of Attraction

The constraints defined in Theorem 1 (see Appendix A) will be used with a Lyapunov to compute the inner approximations corresponding to this set. The entire section is based on the assumption that $V : R^n \rightarrow R$ is a positive and differentiable function satisfying $V(x_0) = 0$ and $V(x) > 0$ when $x \neq x_0$. DA approximations obtained using Lyapunov-based schemes typically use statements like [30], if for positive c and ϵ .

$$V(x) \leq c \Rightarrow \frac{\partial V(x)}{\partial x} f(x) \leq -\epsilon V(x) \quad (1)$$

The following expression defines the set correlated to the LF:

$$\Omega = \{ x \in R^n : V(x) \leq c \} \quad (2)$$

The negative time derivative area is specified as:

$$v = \{ x \in R^n : \dot{V}(x) < 0 \} \cup \{0\} \quad (3)$$

Then, Ω is an approximation of the DA corresponding to the origin, such that $\Omega \subseteq v$.

An unsophisticated initial technique for broadening this analysis requires being disjoint from the area specified by:

$$\Sigma = \{ x \in R^n : \dot{V}(x) \geq 0 \} \quad (4)$$

We try to determine an area specified using a set of smooth "Derivative LF" $\dot{V}(x) : R^n \rightarrow R$ where no failure areas are considered. The points corresponding to $\dot{V}(x) = 0$ are known as "barriers".

Hence, we can determine a solution that is true for all x in the following expression:

$$x \in R^n \in \Omega \cap \Sigma = \{x | V(x) \leq c, \dot{V}(x) \geq 0\} \tag{5}$$

this is a part of the area of attraction and is also positively invariant.

Put differently, these constraints evaluate the “safe set” Σ created using the intersection of functions meeting $\dot{V}(x) = 0$, while the c -sublevel set corresponding to $V(x)$ is the latest DA inner approximation.

The present study is based on a function $V(x)$ that is quadratic with respect to the state x .

$$V(x) = x^T P x, \quad P = P^T \in R^{n \times n}, \quad P > 0 \tag{6}$$

We assume that P has two dimensions, which helps simplify the presentation. The outcomes can be generalised for bigger matrices. It is supposed that:

$$P = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} \tag{7}$$

Then

$$V(x) = p_1 x_1^2 + 2p_2 x_1 x_2 + p_3 x_2^2 \tag{8}$$

This form of a LF is used for specifying a general ellipsoid present in the (x_1, x_2) plane, as described by the equation:

$$\Omega = \{x \in R^n : V(x) = p_1 x_1^2 + 2p_2 x_1 x_2 + p_3 x_2^2 = c, \quad c > 0\} \tag{9}$$

The DA is the region that comprises the convergence of all system trajectories to a particular equilibrium position, as specified using Equation (9).

2.2. Concept Illustration for Enlarging the DA

The concept objective is to determine the guaranteed large value of DA specific to a particular nonlinear polynomial system corresponding to a LF and research the state space using a meta-heuristic technique. This concept is illustrated in Figure 1.

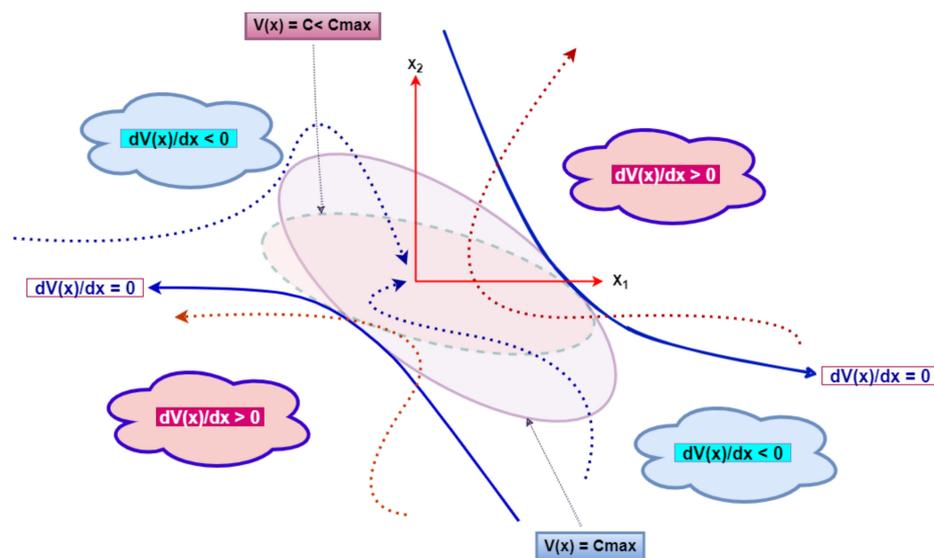


Figure 1. DA concept for a nonlinear system using a QLF.

2.2.1. Research on Maximum DA

As per (9), a greater level set value c corresponds to relatively precise DA estimation. The computation of the LF’s maximum level set relates to DA estimation. It can be determined by solving the pseudo-optimization as specified below:

$$\begin{cases} \max_{c, x} c \\ \text{s.t. } \{x \text{ belong to level set } V(x) - c = 0\} \end{cases} \tag{10a}$$

$$\{x \text{ is any point belonging to region } \Omega \text{ contained in } D\} \tag{10b}$$

The basis of problem (10) is to determine the maximum level set corresponding to $V(x)$ (10a), which lies entirely within the definitive negative region of $\dot{V}(x)$ (10b). It may be obtained by determining the globally optimal solution for the following problem:

$$\begin{cases} \min_{c, x} c \\ \text{s.t. } V(x) - c = 0 \\ \dot{V}(x) = 0 \\ c > 0 \end{cases} \tag{11}$$

The intent behind using problem (11) is to determine the minimum level set corresponding to function $V(x)$, as contained in the level set $\dot{V}(x) = 0$. The required solution is one point located in the state space that correlates to the level sets $V(x) = c$ and $\dot{V}(x) = 0$ coming in contact. The nonlinear nature of model (11) is highlighted; consequently, it might have several local solutions. To be able to converge to an acceptable estimate, global optimality must be attained. This concept is graphically presented using Figure 2. The optimisation problem defined in (11) is nonlinear, and several solutions can be identified. Nevertheless, most solutions are inappropriate for estimating the DA. The following schematic version is assessed for additional evaluation of this concept.

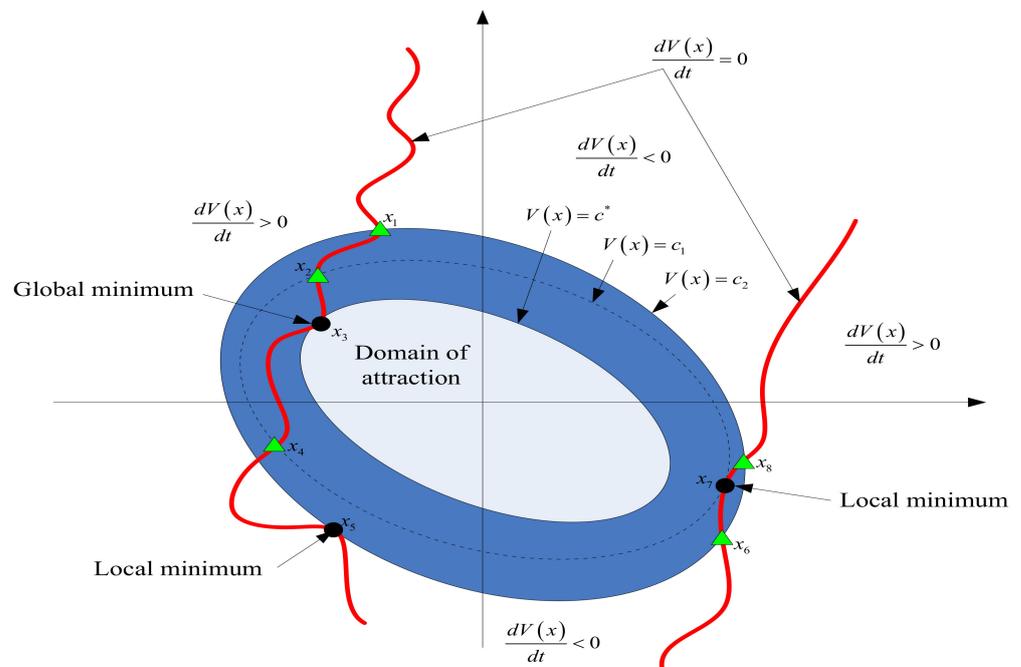


Figure 2. The problem of the global minimum.

Figure 2 presents the global optimization problem qualitatively.

The expression mentioned below indicates a hypersurface:

$$dV(x)/dt = 0, \quad x \neq 0 \tag{12}$$

This is related to the area boundary where $\dot{V}(x)$ is negative definite, where we intend to determine the guaranteed estimation Ω . Such expressions corresponding to QLF denote the inside of the ellipsoid specified using (9).

Distinct points on this figure can be derived as optimization problem solutions (11). The requirements for optimisation (11) apply for points x_5 and x_7 . The LF and its differentiated version verify the sign change corresponding to a particular area in the state space. However, other paths cross the predicted DA transversally. These solutions correspond to local minima and are considered dummy; hence, these are not considered. Furthermore, coordinates $x_1, x_2, x_4, x_6,$ and x_8 validate this inference. Considering the intended DA, several transversal intersections can be found.

Nevertheless, it can be observed that x_3 is considered as a global minimum in Figure 2. It should be noted that all system paths intersect the computed DA tangentially. It is crucial to verify the below-mentioned constraint in order to specify the optimisation problem (11) in a refined way.

Constraint 1:

$$\begin{cases} V(x) = c \\ \dot{V}(x) < 0 \\ c > 0 \end{cases} \tag{13}$$

Constraint 2: The optimal solution must correspond to a state-space area prior to a sign change corresponding to $V(x)$ and its derivative tackles as $|x|$ approaches infinity.

Constraint 3: The level set specified by:

$$V(x) = c \tag{14}$$

corresponds to a global minimum.

It is necessary to ensure that these three constraints are met so that “dummy solutions” can be avoided and the determined globally optimal formulation (11) outputs tangential coordinates. To have detailed examples related to this concept, the reader can consult the following reference [58].

2.2.2. Algorithm of Computing the DA

We now specify algorithmic aspects that enhance DA estimates iteratively using numerous calculations of $V(x)$ that help determine the state x where the first constraint is fulfilled. These cycles enhance the approximation of the radius of the contained ellipse (9).

Before algorithm execution begins, the origin is in an asymptotically stable state. Hence, as specified by the linear theory, there exists a locally applicable QLF:

$$V(x) = x^T P x \tag{15}$$

where P is a positive definite, symmetric matrix having $n \times n$ dimension. The below mentioned Lyapunov expression is solved to compute P .

$$PA + A^T P = -Q \tag{16}$$

corresponding to a positive definite matrix Q having $n \times n$ dimension where A denotes the Jacobian of $f(x)$ at equilibrium.

The derivative of the LF is computed as:

$$\dot{V}(x) = \frac{dV(x)}{dx} f(x) \tag{17}$$

State-space sampling facilitates an instantaneous assessment of $V(x)$ and $\dot{V}(x)$.

This technique changes lower and upper bounds corresponding to c denoted c_{\min} and c_{\max} , respectively. When combined, these values provide a more precise DA estimate. The lower bound c_{\min} is zero, while the upper bound c_{\max} is infinite when the algorithm begins execution. Considering a randomly selected $x^{(i)}$ and conditions $\dot{V}(x^{(i)}) < 0$ and $c_{\min} < V(x^{(i)}) < c_{\max}$ are met, the lower bound is changed to the LF's value, i.e., $c_{\min} = V(x^{(i)})$. On contrast, if $\dot{V}(x^{(i)}) \geq 0$ and $V(x^{(i)}) < c_{\max}$ conditions are met, then the upper bound is changed to $V(x^{(i)})$. When the sampling process has processed numerous samples, the lower bound increases; however, the increase is not always monotonic. The final convergence point c and most significant sublevel set $\Omega(c_{\min})$ is determined. Furthermore, c_{\max} exhibits a monotonic decrease and eventually converges down to c .

When the requirements for Theorem 1 are met corresponding to state $x^{(i)}$, then $V(x^{(i)})$ represents a probable estimate of c therefore, it is retained in an array. This condition ensures that the approximate DA values calculated using c_{\min} satisfy the conditions corresponding to Theorem 1, thereby providing more precise estimates. Array \mathfrak{S} initially contains zero.

Considering that conditions $\dot{V}(x^{(i)}) < 0$ and $V(x^{(i)}) < c_{\max}$ are met, \mathfrak{S} stores the value of $V(x^{(i)})$, i.e., $\Omega(V(x^{(i)}))$, which is a probable DA estimate. If conditions $\dot{V}(x^{(i)}) \geq 0$, $V(x^{(i)}) < c_{\max}$ and $c_{\min} \geq c_{\max}$ are met, then the method tries to select a new lower bound from the values stored in the array. Condition $c_{\min} < c_{\max}$ is considered when trying to identify the maximum value from the array. Choosing an existing lower bound value meets the constraint $\dot{V}(x^{(i)}) < 0$ for the identified sublevel set $\Omega(c_{\min})$. The lower bound is zero, considering the worst-case situation.

The sampling technique used in this study is conservative; however, there could be a substantial overestimation of DA. One example is the region $\dot{V}(x^{(i)}) < 0$ not having a simple connection. Here, the technique might not disregard tiny holes inside the area where $\dot{V}(x^{(i)}) \geq 0$. Presently, there is no assurance that this algorithm always converges; however, empirical data obtained using numerous simulations and real-world tests indicate that this method converges to the precise level set if the sample count is large.

Algorithm 1 presents a series of steps for identifying the DA corresponding to the origin of (A1). Ensuring that Lyapunov theory concepts are met, the following steps are followed:

Algorithm 1 DA estimation

Initialization: Initialize the parameters c , c_{\min} , c_{\max} , $\mathfrak{S} = \{0\}$

Liniarization: The Jacobian of system (A1) at the origin and determine the matrix LF

$$A \leftarrow \partial f / \partial x(x) \Big|_{x=0}$$

$$P \leftarrow \text{solution of } P^T A + AP = -I$$

$$V(x) = x^T P x$$

for $i = 1 : n$ **do**

Evaluate $V(x^{(i)})$, $\dot{V}(x^{(i)})$ for each state

if $(V(x^{(i)}) = c_{\max}) \ \&\& \ (\dot{V}(x^{(i)}) = 0)$

$$\mathfrak{S} = \{V(x^{(i)})\}$$

else if $(V(x^{(i)}) < c_{\max}) \ \&\& \ (\dot{V}(x^{(i)}) \geq 0)$

$$c_{\max} = V(x^{(i)})$$

If $c_{\min} \geq c_{\max}$

$$c_{\max} = \max\{c \in \mathfrak{S}, c \leq c_{\max}\}$$

end

end

end for

Result: Return the best value of DA

Draw the graphic Domain definite by $\Omega = \{V(x^{(i)}) = c_{\max}\}$;

End.

Our technique relies on ascertaining state x in space where DA radius is maximised, corresponding to a particular ellipsoid having boundaries situated within the negative time derivative area. This technique is not feasible for higher dimensions; moreover, state–space discretisation and several simulations would be needed.

3. Proposed Krill Herd Algorithm for State Assessment

3.1. The Motivation for Using the Krill Herd Method

Extensive research has been conducted to determine the mechanisms that result in the formation of non-random patterns by marine animal populations [18]. Several mechanisms have been understood: safety from predators, feeding, environmental characteristics, and better reproduction [23].

The Antarctic krill is one of the most commonly researched marine species [24]. In fact, the krill herd is characterized by several uncertainties regarding its representative distribution [25]. Various conceptual frameworks have been proposed to explain the pattern of krill herds [26]. The results suggest that krill swarms are the fundamental organizational unit of this species.

Marine predators such as sea birds or penguins attack krill by leading individual krill to an area with lesser krill density. Following the predatory attack, krill herd formation has two primary objectives: (1) enhance krill density and (2) access to food. Krill behaviour to enhance the density and locate food are identified as the objective function. Subsequently, herding is observed around local minima. Individual krill movement is such that the best solution can be found in this search for food and increased density.

3.2. Krill Herd

Gandomi and Alavi first proposed the Krill Herd (KH) algorithm in 2012. KH [27] is a novel meta-heuristic technique for addressing optimisation problems. Krill swarm patterns during food search are the motivation for this algorithm. The position of individual krill is affected by three aspects specified below [27]:

- i. Krill movement due to other individuals;
- ii. Foraging behaviour;
- iii. Random diffusion;

A simple Lagrangian model [27] can be formulated for the three aspects of KH. Equation (18) specifies this model:

$$\frac{dX_k}{dt} = N_k + F_k + D_k \quad (18)$$

where

N_k : krill motion due to other individuals;

F_k : foraging-specific movement;

D_k : is the physical diffusion corresponding to the k^{th} individual krill;

k : denotes a single krill;

t : generation count;

3.2.1. Krill Movement Due to Other Individuals

The first movement direction is approximated using three characteristics: repulsive effect, target effect, and local effect. The movement of an individual krill i^{th} is modelled as:

$$N_k^{next} = N^{\max} \alpha_k + \omega_d N_k^{present} \quad (19)$$

where

$$\alpha_k = \alpha_k^{local} + \alpha_k^{target} \tag{20}$$

and

N_k^{max} denotes the maximum induced speed,
 ω_d denotes inertia weight in the range [0,1],
 $N_k^{present}$ denotes the previous induced motion of the i^{th} individual krill
 α_k^{target} and α_k^{local} denote the target and local effects, respectively.

3.2.2. Foraging Movement

Two primary aspects determine the second movement: location of food, and prior experience concerning the location of food. This motion can be modelled for the i^{th} krill as specified below [27]:

$$F_k^{next} = V_f \beta_k + \omega_f F_k^{previous} \tag{21}$$

where

$$\beta_k = \beta_k^{food} + \beta_k^{best} \tag{22}$$

and

V_f denotes foraging speed; for the present study, V_f is set to 0.02 [27].
 ω_f denotes inertia weight in the range [0,1],
 $F_k^{previous}$ denotes previous foraging movement,
 β_k^{food} denotes food attraction
 β_k^{best} is the influence of best fitness

3.2.3. Random Diffusion

The third motion is fundamentally random. Two factors are used for expressing the model corresponding to this motion: random vector and highest diffusion speed. The model expression is specified below [27]:

$$D_k = D^{max} \delta V_f \tag{23}$$

where,

D^{max} is the maximum induced speed;
 δ denotes the random directional vector [0,1]

3.3. Movement in KH

Typically, the motions specified in KH change often with respect to krill position to attain the best position. The two subsequent motions (2nd and 3rd) caused by other krill comprise two local and two global techniques used concurrently to make KH an efficacious and potent technique. The following equations specify the location of individual krill i in the $[t, t + \varepsilon]$ time interval:

$$X_k(t + \varepsilon) = X_k(t) + \varepsilon \frac{dX_k}{dt} \tag{24}$$

where ε is a crucial constant (speed vector scaling factor).

Equation (25) may be used to determine the value of ε

$$\varepsilon = P_{cst} \sum_{j=1}^{N_v} (U_j - L_j) \tag{25}$$

where,

N_v : denotes the total variable count
 U_j, L_j : denotes the upper and lower bounds corresponding to the j th variables
 P_{cst} : position constant in the [0,2] range.

To achieve better optimisation and enhanced convergence speed, multiplication and crossover aspects specific to the algorithm are integrated with KH [27].

3.3.1. Crossover

The crossover probability (Cr) parameter regulates the crossover process. The position of every individual krill is changed based on its interaction with other krill. The j th component corresponding to the i th krill may be specified as [27]:

$$X_{i,j} = \begin{cases} X_{m,j} & \text{if } rand \leq Cr \\ X_{i,j} & \text{if } rand > Cr \end{cases} \quad (26)$$

where, $m = 1, 2, \dots, N$ and N denotes krill population and z_i^{best} : is the best historical meeting point for krill i .

3.3.2. Mutation

The mutation probability (Mr) parameter regulates the mutation process; it may be specified as [27]:

$$X_{i,j} = X_{best,j} + \theta(X_{s,j} - X_{t,j}) \quad (27)$$

where,

$X_{best,j}$: globally best vector,

$X_{s,j}, X_{t,j}$: two random vectors,

θ : scalar in the [0,1] range.

The updated $X_{i,j}$ value is expressed as [27]:

$$X_{i,j}^{mod} = \begin{cases} X_{i,j}^{new} & \text{if } rand \leq Mr \\ X_{i,j} & \text{if } rand > Mr \end{cases} \quad (28)$$

3.4. Computation of the Fitness Function

Execution of the KH algorithm comprises the determination of the fitness function as its primary step. Correspondingly, particle quality assessment is made using the domain of attraction. $\Omega = \{x \in R^n : V(x^{(i)}) \leq c\}$, $c > 0$ is the objective function expression that must be maximised. Considering the i^{th} particle position $x^{(i)}$, the third step of Algorithm 2 needs to be verified. States $x^{(i)}$ and the domain of attraction $\Omega(V(x^{(i)}))$ can be used as per the below-mentioned process to determine the fitness function corresponding to the i th particle.

3.5. KH Computation Procedure

Algorithm 2 lists the computational steps for this process.

3.6. Fundamentals of Chaotic KH (CKH)

All metaheuristic algorithms share a similar search scheme divided into two parts: exploration and exploitation. During the exploration phase, the search space is evaluated extensively to generate additional solutions. In contrast, the exploitation phase comprises a rapid convergence of the algorithm to the global optimum. These two steps are sophisticated aspects required to determine the global optimum; here, an accurate solution is typically balanced between these two factors. However, a significantly complex optimisation such as a DA problem with an unknown search space presents challenges. Hence, the most appropriate trade-off between exploitation and exploration cannot be ascertained. In order to focus on these two aspects in the search space, we have considered several adaptive and random variables along with the KH.

Algorithm 2 The KH computational process**Begin**

Initialization. Set the initial values for the population size (N_p), the maximum induced speed (N_{obj}^{max}), the foraging speed (N_{obj}^{max}), the maximum diffusion speed (N_{obj}^{max}), the generation counter (m) and the maximum number of fitness function evaluations (N_{obj}^{max}).

Fitness evaluation. Evaluate the fitness function and randomly set up the position $x^{(i)}$; $i = 1, 2, \dots, N_p$ set of each krill individual then assess the fitness function value for all individuals.

While $m < (N_{obj}^{max})$ or the stopping criteria are not accomplished **do**

Organize the population initiating from the best to worst.

Keep the best krill individual in "STORE"

for $i = N_p$ **do**

Compute the:

- a) Physical diffusion
- b) Foraging motion
- c) Induced motion

Implement genetic operators

Update the krill individual position in the search domain

Assess each krill individual based on its updated position

end for

Organize the population from best to worst and fix the current best one. Increment the counter m

$m = m + 1$

end while

Results: Presentation of the results and visualization;

End.

Chaos is among the mathematical techniques that have recently been used for enhancing exploitation and exploration [27]. Chaos theory provides the leading method for enhancing evolutionary algorithm performance concerning the ignorance of local optima and enhancing convergence rate; moreover, the chaos technique does not rely on random parameters. Additionally, except for chaotic maps, another fundamental enhancement is the integration of the elitism approach to the CKH framework. Like other optimisation schemes, we add elitism in some form to retain the optimal krill population.

The elitism scheme which is implemented in the CKH algorithm is considered as one of the most important enhancements offered by this. As with peer population-based optimization strategies, some kinds of elitism are typically combined so as to retain the best solutions in the investigated population. In the main cycle of the CKH, to start with, the KEEP best population results are saved in a variable KEEPKRILL. Generally, the KEEP worst solutions are eventually replaced by the KEEP best solutions. This later elitism process can ensure that the whole population cannot be declined to the population with worse fitness than the former. This avoids the best population from being abandoned by the dynamic of the motion calculation operator. Note that the elitism strategy is used to save the feature of the krill that has the best fitness in the CKH process, so even though motion calculation operation degrades its matching krill, this later has been stored and can be recovered by its previous good status if required.

Inertia weights ω_d , ω_f are the primary parameters that regulate the convergence rate of KH. To enhance search efficacy and ensure convergence to the best solution, the chaos concept is integrated with the KH framework for this study's scope. Considering that Chebyshev maps are the most-extensively used chaotic behavioural maps, it is possible to generate chaotic sequences rapidly and efficiently. Furthermore, there is no need to retain long sequences. Chebyshev maps used during the CKH method change the value of randomised parameters $\omega^{chebychev} = (\omega_d, \omega_f)$ in KH.

Chebyshev maps update parameters ω_d and ω_f in accordance with the Equations (19) and (21):

$$\omega_j^{chebychev} = \cos\left(j * \cos^{-1}\left(\omega_{j-1}^{chebychev}\right)\right) \tag{29}$$

Equation (29) produces a chaotic sequence in the [0,1] range. For every independent execution of equation (29), $\omega_0^{chebychev}$ is output randomly. The chaotic value $\omega_j^{chebychev}$ produced using a logistical map with 300 runs and $\omega_0^{chebychev} = 0.001$ is depicted in Figure 3.

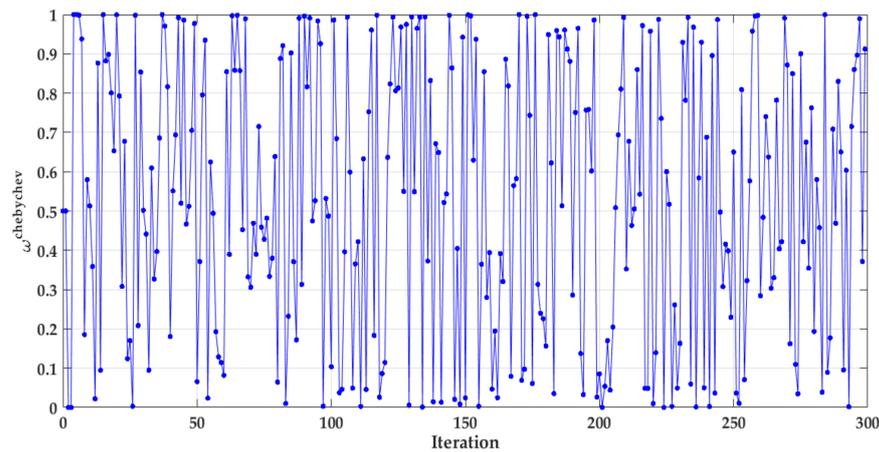


Figure 3. Weights generated $\omega^{chebychev} = (\omega_d, \omega_f)$ using a Chebychev chaotic map over 300 iterations.

In the case of CKH, Equations (19) and (21) can be written as:

$$N_k^{next} = N^{max} \alpha_k + \omega_{d,j}^{chebychev} N_k^{present} \tag{30}$$

$$F_k^{next} = V_f \beta_k + \omega_{f,j}^{chebychev} F_k^{previous} \tag{31}$$

Algorithm 3 presents the pseudocode for the CKH computation process.

Algorithm 3 The CKH computational process

Begin

Initialization. Initialize the CKH’s parameters $m, (N_p), (N_{obj}^{max}), (N_{obj}^{max}), (N_{obj}^{max})$ and (N_{obj}^{max}) . Set up the chaotic maps value in random way and inertia weights $\omega^{chebychev} = (\omega_d, \omega_f)$ as well. STORE: number of the best krill swarm to retain from one generation to the next.

Fitness evaluation. Evaluate the fitness function and randomly set up the position $x^{(i)}; i = 1, 2, \dots, N_p$ set of each krill individual, then assess the fitness function value for all individuals.

While $m < N_{obj}^{max}$ or the stopping criteria are not accomplished **do**

Organize the population initiating from the best to worst.

Keep the best krill individual in “STORE”

for $i = 1: N_p$ **do**

Compute the:

- a) Physical diffusion
- b) Foraging motion
- c) Induced motion

Implement genetic operators

Update the krill individual position in the search domain

Assess each krill individual based on its updated position

end for

Update the “STORE” with the new best value;

Organize the population from best to worst and fix the current best one Increment the

counter $m: m = m + 1$

end while
Results: Presentation of the results and visualition;
End.

The flowchart describing the developed strategy is described in Figure 4.

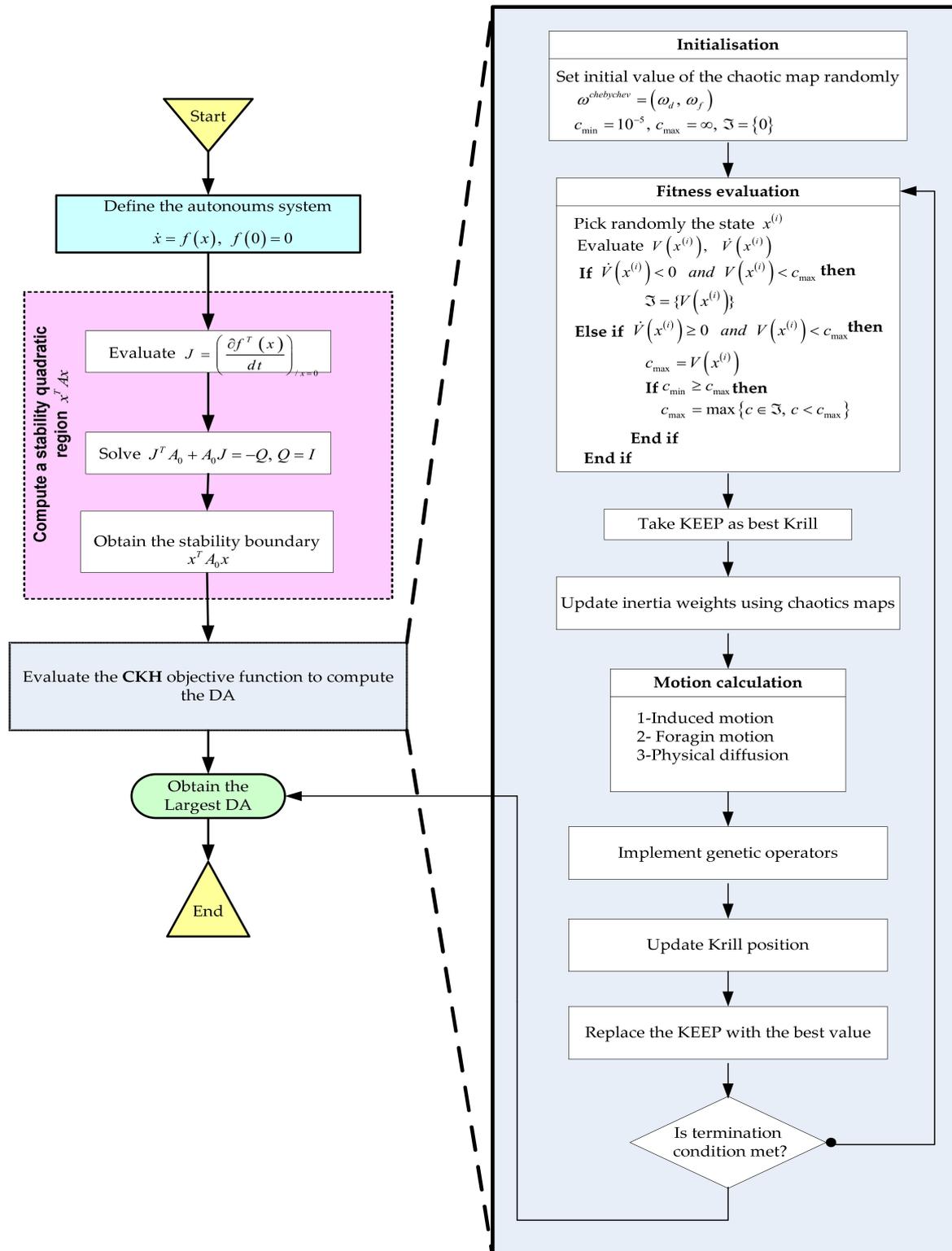


Figure 4. Flowchart of the CKH algorithm applied for the DA estimation problem.

4. Illustrative Examples

4.1. Example 1

The nonlinear dynamic expression below might be assumed as a model for a pendulum [28].

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\sin(x_1) - 0.5x_2 \end{cases} \quad (32)$$

where the pendulum, having angular velocity x_2 , is at an angle x_1 with the vertical axis. The state vector is specified as: $x = [x_1, x_2]^T$. We use the proposed approach with several chaotic map distributions to find an approximate DA corresponding to the stable equilibrium $x = (0, 0)$. To calculate a potential LF, we linearise the dynamic model for a pendulum. Hence, the objective is to identify an acceptable stability domain:

$$\Omega = \{x \in R^n : V(x) \leq c\}, \quad c > 0 \quad (33)$$

where c is a real non-zero constant and $V(x)$ denotes the positive definite QLF expression specified below:

$$V(x) = x^T P x, \quad P^T = P \quad (34)$$

P denotes a symmetric and positive matrix that is ascertained by determining the solution of the Lyapunov expression.

$$A^T P + P A = -Q \quad (35)$$

Q is an identity matrix as specified below:

$$Q = I \quad (36)$$

In this case,

$$P = \begin{bmatrix} 2.25 & 0.5 \\ 0.5 & 1 \end{bmatrix} \quad (37)$$

The computed LF is:

$$V(x) = x^T \begin{bmatrix} 2.25 & 0.5 \\ 0.5 & 1 \end{bmatrix} x \quad (38)$$

which can be expressed as:

$$V(x) = 2.25x_1^2 + x_1x_2 + 2x_2^2 \quad (39)$$

In the case the derivative of $V(x)$, as defined by the expression below, is negative definite, and the nonlinear model has assured stability.

$$\dot{V}(x) = \dot{x}^T P x + x^T P \dot{x} \quad (40)$$

This equation may be specified as:

$$\dot{V}(x) = 4x_1x_2 - x_2^2 - x_1 \sin(x_1) - 4x_2 \sin(x_1) \quad (41)$$

In this regard, we start encoding variables where particle position is defined according to the below-mentioned vector:

$$x^{(i)} = \begin{pmatrix} x_1^{(i)} \\ x_2^{(i)} \end{pmatrix}; \quad i = 1, \dots, N_p \quad (42)$$

Initially, we consider that the LF has quadratic nature.

Consider

$$V(x^{(i)}) = 2.25x_1^{2(i)} + x_1^{(i)}x_2^{(i)} + 2x_2^{2(i)}; i = 1, \dots, N_p \tag{43}$$

whose time derivate is specified below.

$$\dot{V}(x^{(i)}) = 4x_1^{(i)}x_2^{(i)} - x_2^{2(i)} - x_1^{(i)}\sin(x_1^{(i)}) - 4x_2^{(i)}\sin(x_1^{(i)}); i = 1, \dots, N_p \tag{44}$$

Fitness value can be determined using maximum $\dot{V}(x) = 4x_1x_2 - x_2^2 - x_1\sin(x_1) - 4x_2\sin(x_1)$ for which the algorithm has an acceptable solution, as mentioned in the pseudo-code specified in Section 3.

Subsequently, the proposed approach is used to ascertain $V(x^{(i)})$, $\dot{V}(x^{(i)})$ and c .

The settings used for the CKH technique are specified as follows. The scenario is based on a population size of 25, maximum diffusion speed of 0.005, foraging speed of 0.02, running for 150 iterations and a krill number of 25. The CKH algorithm was executed 25 times, and evolution was recorded in Figure 5a–d. These figures depict how the objective function evolved. After evaluating the optimisation outcome, the following was selected:

$$x^{(i)} = \begin{pmatrix} x_1^{(i)} \\ x_2^{(i)} \end{pmatrix} = \begin{pmatrix} 1.746 \\ 0.747 \end{pmatrix} \tag{45}$$

while the domain is:

$$\Omega = \{x \in R^n : V(x) = 2.25x_1^2 + x_1x_2 + 2x_2^2 = 9.2870\} \tag{46}$$

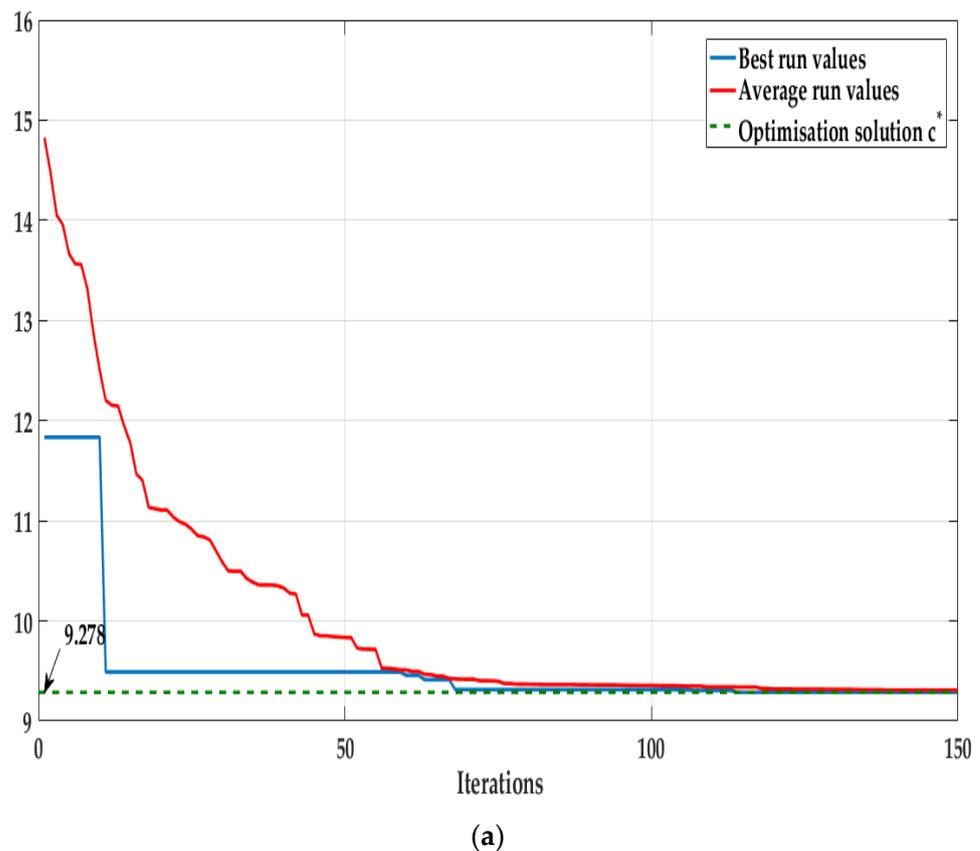


Figure 5. Cont.

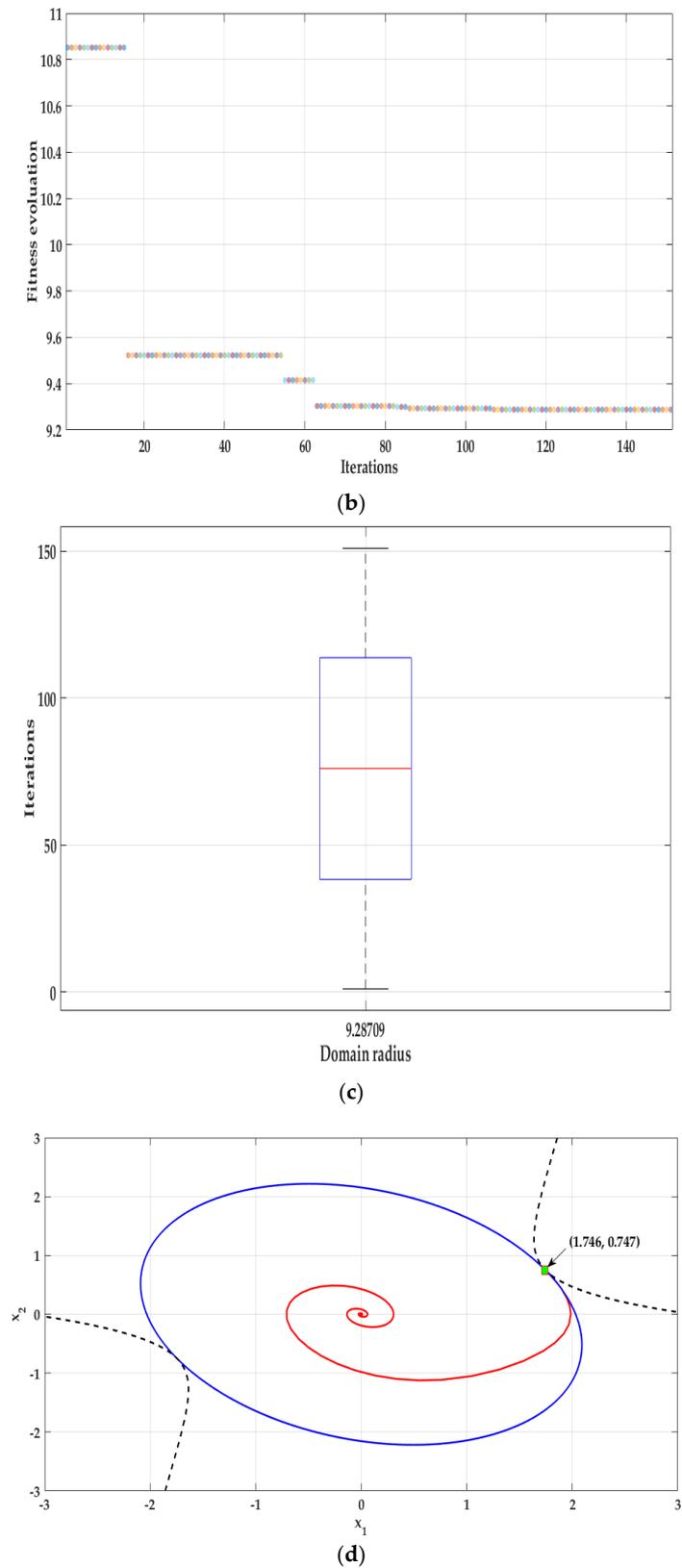


Figure 5. (a) Evolution of DA optimization process using CKH corresponding to example 1. (b) Swarm plot diagrams using CKH corresponding to example 1. (c) Box diagram using CKH corresponding to example 1. (d) Ellipsoid and phase trajectory that originates from the tangency point on the largest ellipsoid.

The dynamic of the state variables initialized from the tangency point state locus is represented in Figure 6. This is just to confirm the reliability of the obtained solution; indeed, the state variables converges asymptotically to the stable equilibrium point.

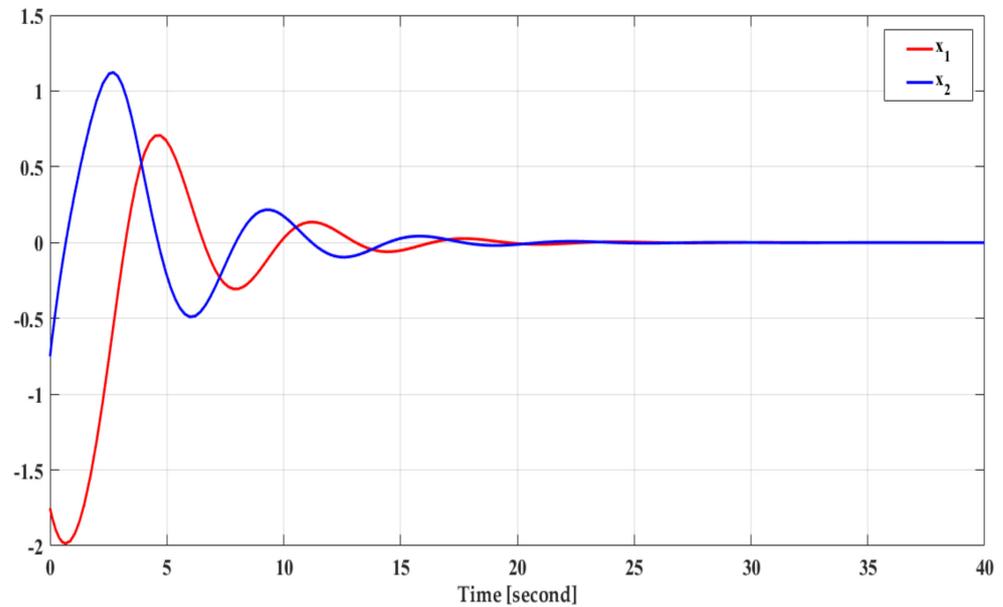


Figure 6. State variables dynamics initialized on the tangency point.

4.2. Example 2

The below-mentioned expression is the state-space equivalent of the Van der Pool equation [28].

$$\begin{cases} \dot{x}_1 = -x_2 \\ \dot{x}_2 = x_1 - x_2 + x_1^2 x_2 \end{cases} \tag{47}$$

Here, $x = [x_1, x_2]^T$ denotes the state vector. We use our approach comprising several chaotic maps for estimating the general DA corresponding to a stable equilibrium $x = (0, 0)$. Initially, the dynamic model of Van der Pol is linearised in proximity to the equilibrium to determine a potential LF.

$$A = \left. \frac{\partial f}{\partial x} \right|_{(0,0)} = \begin{bmatrix} 0 & -1 \\ 1 & -1 \end{bmatrix} \tag{48}$$

The symmetric matrix P is positive definite, and it is computed by solving the Lyapunov expression:

$$A^T P + PA = -Q \tag{49}$$

Q is an identity matrix.

$$Q = I \tag{50}$$

In this case,

$$P = \begin{bmatrix} 1.5 & -0.5 \\ -0.5 & 1 \end{bmatrix} \tag{51}$$

The potential LF is specified as:

$$V(x) = x^T \begin{bmatrix} 1.5 & -0.5 \\ -0.5 & 1 \end{bmatrix} x \tag{52}$$

which can be expressed as:

$$V(x) = 1.5x_1^2 - x_1x_2 + x_2^2 \tag{53}$$

The stable region corresponding to the nonlinear model is assured if the derivate of $V(x)$

$$\dot{V}(x) = \dot{x}^T Px + x^T P \dot{x} \tag{54}$$

is negative definite.

$$\dot{V}(x) = -x_1^2 - x_2^2 - x_1^3 x_2 + 2x_1^2 x_2^2 \tag{55}$$

In this regard, the following vector is initially employed for variable encoding related to particle position.

$$x^{(i)} = \begin{pmatrix} x_1^{(i)} \\ x_2^{(i)} \end{pmatrix}; i = 1, \dots, N_p \tag{56}$$

We initially assume that the LF is quadratic.

Let

$$V(x^{(i)}) = 1.5x_1^{2(i)} - x_1^{(i)}x_2^{(i)} + x_2^{2(i)}; i = 1, \dots, N_p \tag{57}$$

Then its time derivative can be specified as:

$$\dot{V}(x^{(i)}) = -x_1^{2(i)} - x_2^{2(i)} - x_1^{3(i)}x_2^{(i)} + 2x_1^{2(i)}x_2^{2(i)}; i = 1, \dots, N_p \tag{58}$$

Fitness value can be specified using the maximum c^* where a feasible solution exists for the algorithm specified as pseudocode in Section 3.

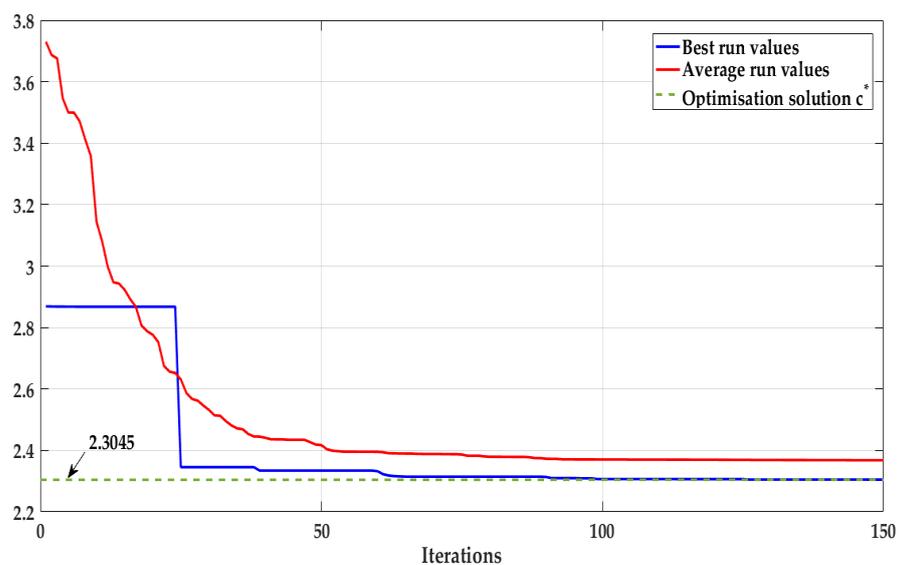
Subsequently, use the proposed approach to determine $V(x^{(i)})$, $\dot{V}(x^{(i)})$ and c .

The CKH algorithm was configured as: 0.02 foraging speed, 150 iteration count, 0.005 maximum diffusion speed, 25 population size. The CKH algorithm was executed 20 times. Objective function evolution history is specified in Figure 7a and the DA show in Figure 7b. Considering the optimisation outcomes, we identify:

$$x^{(i)} = \begin{pmatrix} x_1^{(i)} \\ x_2^{(i)} \end{pmatrix} = \begin{pmatrix} -0.8538 \\ 0.7534 \end{pmatrix} \tag{59}$$

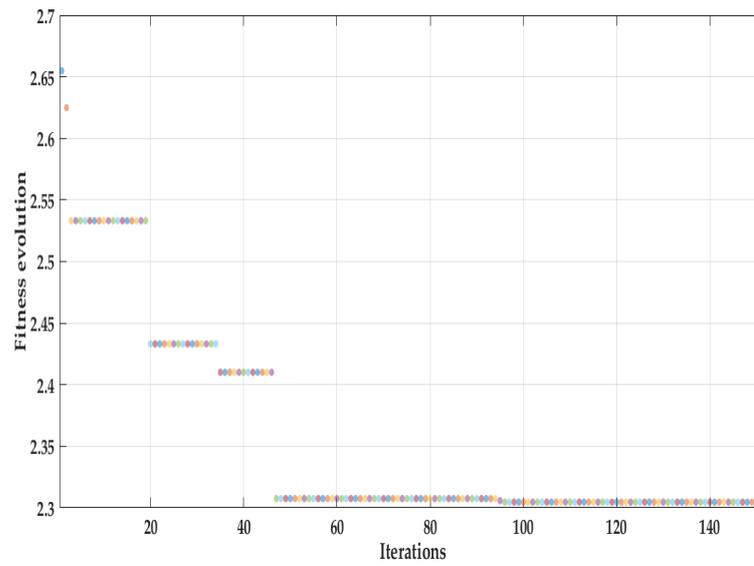
while the more extensive domain is:

$$\Omega = \{x \in R^n : V(x) = 1.5x_1^2 - x_1x_2 + x_2^2 = 2.3045\} \tag{60}$$

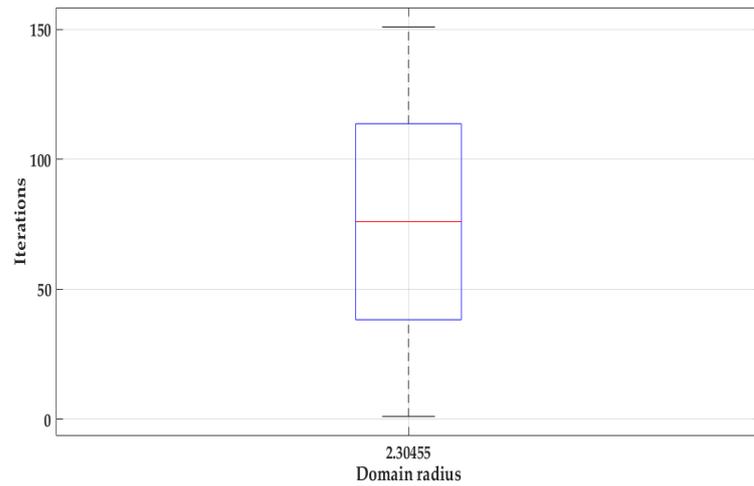


(a)

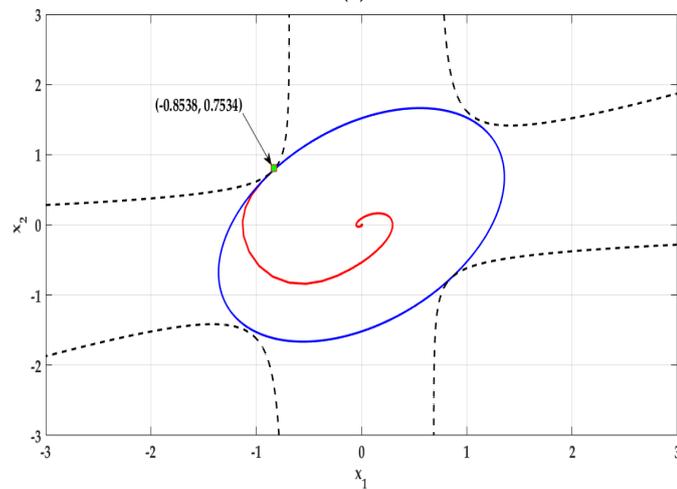
Figure 7. Cont.



(b)



(c)



(d)

Figure 7. (a) Evolution of DA optimization process using CKH corresponding to example 2. (b) Swarm plot diagrams using CKH corresponding to example 2. (c) Box diagram using CKH corresponding to example 2. (d) Ellipsoid and phase trajectory that originates from the tangency point on the largest ellipsoid.

The dynamic of the state variables initialized from the tangency point state locus is represented in Figure 8. This is just to confirm the reliability of the obtained solution; indeed, the state variables converges asymptotically to the stable equilibrium point.

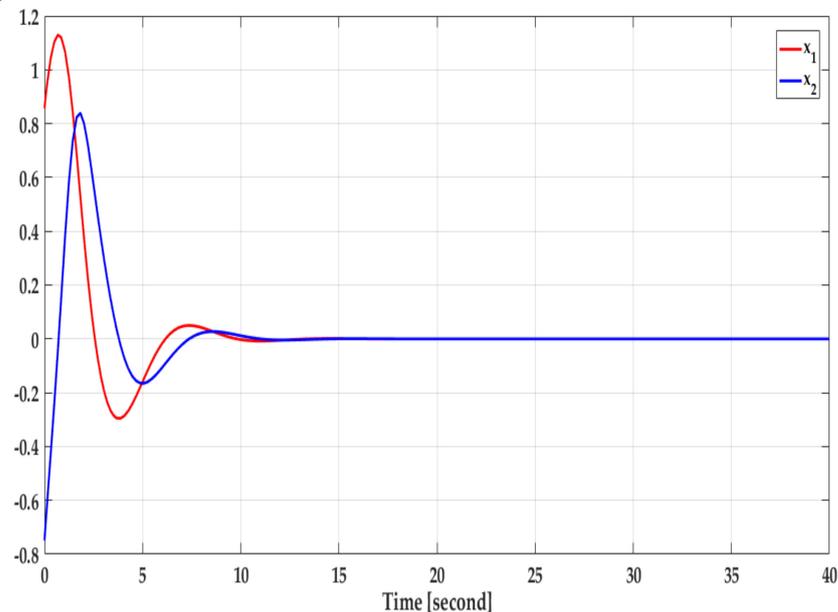


Figure 8. State variable dynamics initialized on the tangency point.

Unlike existing techniques, the developed method can be applied for both polynomial and rational LF. Indeed, it allows the development of a chaotic operative optimization algorithm which ensures the convergence to an enlarged DA in a significantly limited running time. Despite the performance, offering a larger accurate DA at high speed converging time, CKH is valuable for real-time trajectory tracking control problems

5. Discussion

This section comprises an in-depth discussion of a comparative analysis and assessment of the formulated CKH technique and peer-reviewed methods [28,29].

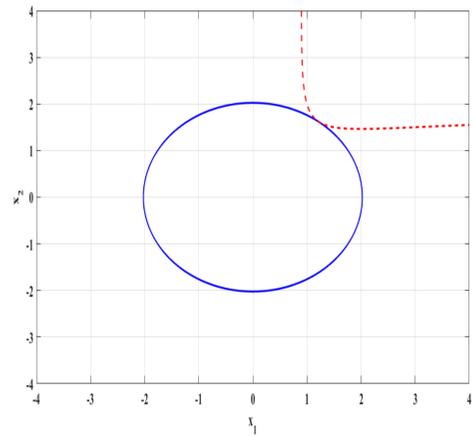
Table 1 lists the approximate DA features corresponding to six dynamic nonlinear benchmark models having a quadratic LF, computed using the literature [28,29]. The nonlinear polynomial and nonpolynomial variants of the investigated examples are presented. E1, E2, E3 and E4 are second-order systems. Nevertheless, E5 and E6 are nonlinear polynomial and nonpolynomial third-order models. The DA remains the primary assessment criteria. The observed outcomes are better, concerning DA values. Figure 9 shows the approximated DAs for the origins in examples E1–E6 described in Table 1 using the CKH method.

Furthermore, the formulated technique offers a comprehensive solution that comprises the definition of a LF, concluding with an enhanced DA. This outcome is obtained for every example assessed in this study. Implementation was straightforward and less time-intensive.

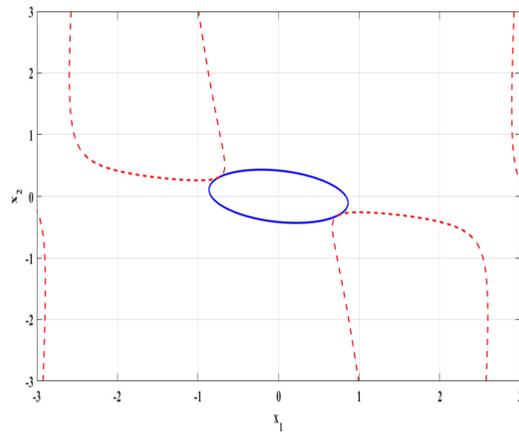
Every example uses the CKH technique to maximise the permissible domain value. These values are contrasted against the results identified using a peer optimisation-specific method mentioned in the literature [28,29]. It is evident from Table 1 that the six assessed samples yield better domains of attraction when identified using the CKH method as compared to optimisation-based techniques. Formulation (11) comprises nonlinear optimisation for a problem with numerous solutions having improper DA estimations corresponding to the system being studied. For instance, consider the many solutions of problem (11) corresponding to systems in E1 and E2, as indicated in Figures 10 and 11.

Table 1. Estimated DA features for CKH, ref. [28,29] methods.

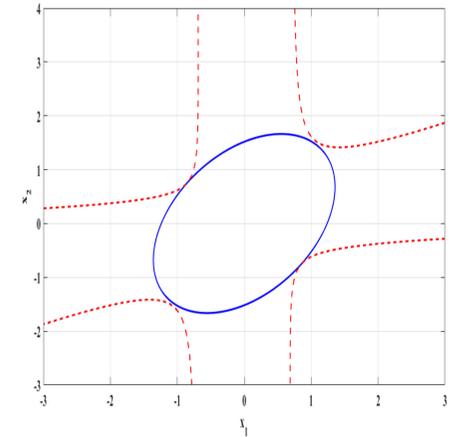
Example	Systems Dynamics	Lyapunov Function	Estimated DA			Optimization Algorithm Convergence Time (ms)			Tangency Point (Best Solution)		
			[29]	[28]	CKH	[29]	[28]	CKH	[29]	[28]	CKH
E1	$\begin{cases} \dot{x}_1 = -2x_1 + x_1x_2 \\ \dot{x}_2 = -x_2 + x_1x_2 \end{cases}$	$V(x) = x_1^2 + x_2^2$	4.0804	4.112	4.0955	-	6.6	0.350	-	-	$\begin{cases} x_1 = 1.223 \\ x_2 = 1.611 \end{cases}$
E2	$\begin{cases} \dot{x}_1 = -x_2 \\ \dot{x}_2 = x_1 - x_2 + x_1^2x_2 \end{cases}$	$V(x) = 1.5x_1^2 - x_1x_2 + x_2^2$	2.09	2.318	2.3045	-	6.7	0.542	-	-	$\begin{cases} x_1 = 0.852 \\ x_2 = -0.755 \end{cases}$
E3	$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -0.2x_2 + 0.81 \sin(x_1) \cos(x_1) - \sin(x_1) \end{cases}$	$V(x) = x_1^2 + x_1x_2 + 4x_2^2$	0.699	0.708	0.6993	-	7.2	0.236	-	-	$\begin{cases} x_1 = 0.741 \\ x_2 = -0.307 \end{cases}$
E4	$\begin{cases} \dot{x}_1 = -0.25x_1 + \ln(1 + x_2) \\ \dot{x}_2 = -0.3750x_1 - 0.2x_1x_2 + (0.125x_1 - x_2) \cos(x_1) \end{cases}$	$V(x) = x_1^2 + x_2^2$	0.2737	0.278	0.2737	-	8.3	0.258	-	-	$\begin{cases} x_1 = 0.443 \\ x_2 = -0.277 \end{cases}$
E5	$\begin{cases} \dot{x}_1 = -x_1 + x_2x_3^2 \\ \dot{x}_2 = -x_2 + x_1x_2 \\ \dot{x}_3 = -x_3 \end{cases}$	$V(x) = x_1^2 + x_2^2 + x_3^2$	4.9188	4.971	4.969	-	8.4	0.77	-	-	$\begin{cases} x_1 = 1.200 \\ x_2 = 1.452 \\ x_3 = -1.190 \end{cases}$
E6	$\begin{cases} \dot{x}_1 = 1 + x_3 + 0.125x_3^2 - e^{(x_1)} \\ \dot{x}_2 = -x_2 - x_3 \\ \dot{x}_3 = -x_2 - 2x_3 - 0.5x_1^2 \end{cases}$	$V(x) = x_1^2 + x_2^2 + x_3^2$	2.655	2.887	2.6617	-	8.6	0.813	-	-	$\begin{cases} x_1 = -1.325 \\ x_2 = 0.611 \\ x_3 = -0.728 \end{cases}$



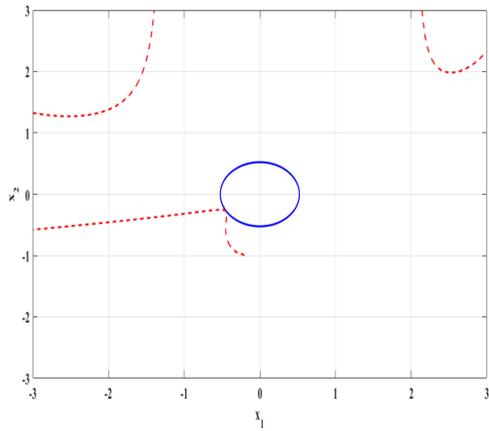
(a) System E1



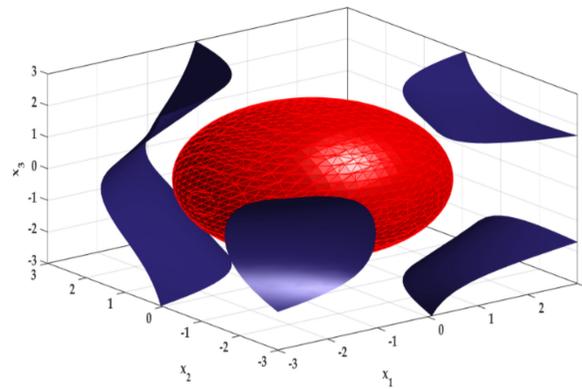
(b) System E2



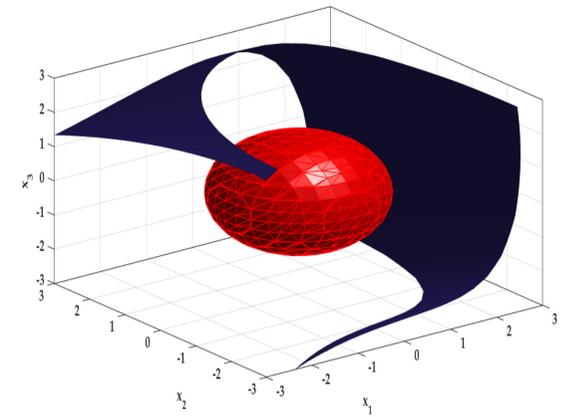
(c) System E3



(d) System E4



(e) System E5



(f) System E6

Figure 9. Approximated DAs for the origins in examples E1–E6 described in Table 1 using the CKH method.

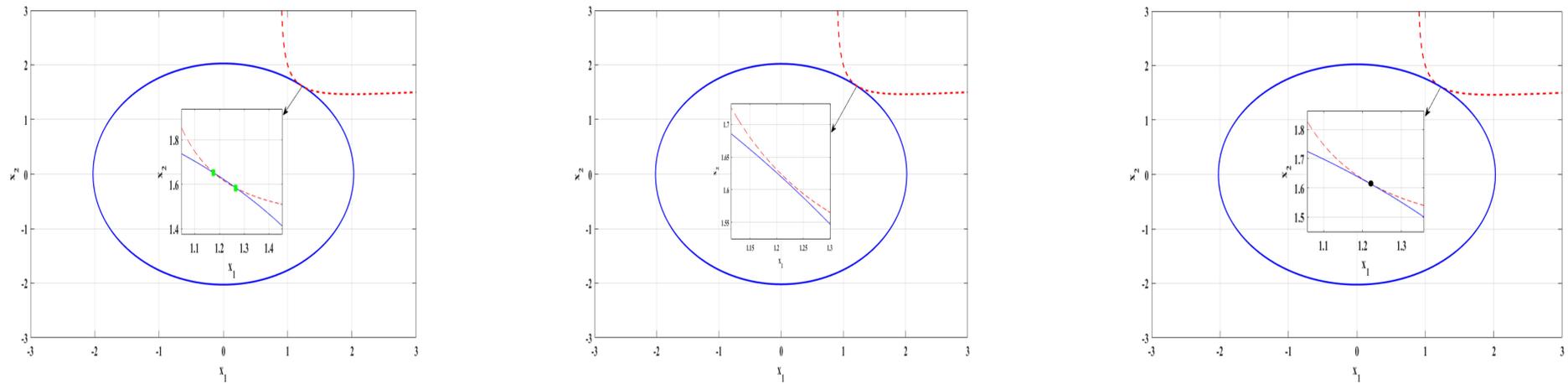


Figure 10. Approximated DAs for the origins of examples E1 described in Table 1 using the CKH method. (a) DA using the sampling method [28]. (b) DA using the LMI method [29]. (c) DA using CKH method.

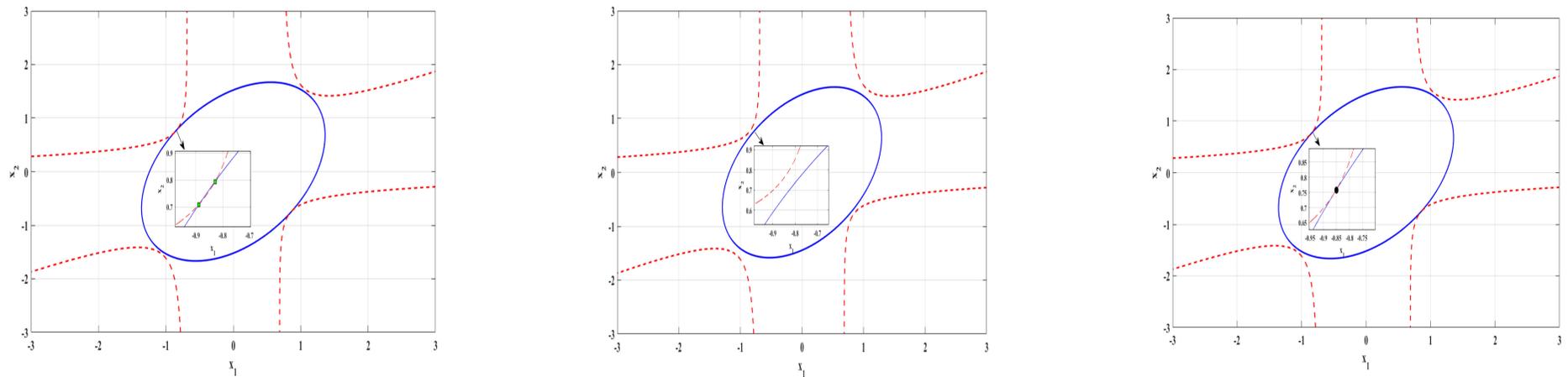


Figure 11. Approximated DAs for the origins of examples E2 described in Table 1 using the CKH method. (a) DA using the sampling method [28]. (b) DA using the LMI method [29]. (c) DA using CKH method.

Figures 10 and 11 depict LF level sets that are indicated using a solid line. Dashed lines are used to depict the time derivative level set at level zero. It is also noteworthy that, like the previous instance, E1 has a unique solution corresponding to $V(x^{(i)}) = c$. The discussion ahead specifies a unique solution which does not satisfy the tangency constraint for the method described in [28]. The later weakness is well clarified in Figure 10a where a clear intersection between the LF and its derivative is shown. Consequently the value of solution $V(x^{(i)}) = c = 4.112$ provided in Table 1 is not realistic. The solution corresponding to E2 solves the problem (11), where c is less than the required estimate in Figure 11a. This solution correlates to a transversal intersection between level sets $V(x^{(i)}) = c$ and $\dot{V}(x^{(i)}) = 0$. It may be observed that these level sets intersect at a significant distance from the origin. Hence, a minor fraction of the level set $V(x^{(i)})$ provides an actual DA estimate (the small circle close to the origin).

For the Van der Pool model, all techniques specified in the literature provide different results. Using the proposed method, we found $V(x) = 2.3045$, while for the technique in [28], the corresponding value is 2.318, which corresponds to an explicit intersection between the LF and its derivative and should be rejected as a solution. At the same time, Hachiho's technique yields a value of 2.09. All these methods were assessed using identical LF: $V(x) = 1.5x_1^2 - x_1x_2 + x_2^2$. Figure 11 depicts the numerous solutions corresponding to the system (A1). The solid line indicates the distinct levels sets corresponding to the LFs. The time derivative for zero value is also depicted using dashed lines.

The solution presented in Figure 11a addresses the problem (11), with c more than the required estimate. When the solution point is determined, two solution points are observed specific to the transversal intersection of level sets described by $V(x) = 2.318$ and $\dot{V}(x) = 0$. Nevertheless, neither of these two are precise DA estimates. Considering that all solutions depicted in Figure 11a are applicable to the problem (11), but none are appropriate, these solutions are therefore considered "dummy".

In Figure 11c, we have indicated that $V(x) = 2.09$ and level set $\dot{V}(x) = 0$ do not intersect. The solution presented in the figure corresponds to the problem (11) having a c value less than the required DA estimation. Nevertheless, DA estimation size is insufficient since the LF level set and that corresponding to its derivative do not intersect.

Considering Figure 11b, it can be observed that the LF intersects its derivative tangentially at two coordinates owing to symmetry. One result is reported because the two solutions are similar concerning DA estimation.

An important observation related to the time cost shown by the compared methods involves the superiority of the CKH hybrid method developed in this paper. Indeed, in [29] no further data have been reported regarding the time cost. In [28] authors reported more than 5 ms time cost overall for the studied examples. It is interesting to notice that CKH involves less than 0.5 ms as a consuming time in the overall investigated example. This is to emphasize the fact that CKH is the better and more appropriate technique to be used in online trajectory tracking problems. Table 2 summarises qualitatively the analytical technique given in [29], the sampling method given in [28] and the CKH optimizing method developed in this work. The comparison criterion focuses mainly on the ability to implement the related algorithms easily and to avoid local optimizing solutions, while ensuring convergence [24].

Table 2. Summary comparative key performance of the investigated methods [28,29] in contrast with the designed CKH method.

Method	Pros	Cons
CKH	<ul style="list-style-type: none"> - High accuracy in defining the solution characterizing the tangency point coordinates and the DA radius. - Reliable meta-heuristic technique, high convergence speed due to the induced chaotic movement parameter with strong robustness. - Global enhanced search ability. - Local optimum avoidance. - Dummy DA solution avoidance. - No random space research is needed. - Interesting feasibility for a wider range of benchmark models. - Performs easily and simply, and no need of parameter tuning. - Efficient for both polynomial and rational function - Reasonable amount of computational requirements allowing to solve control tracking problems. - Smaller number of parameters 	<ul style="list-style-type: none"> - Suitable chaotic map should be selected. - Suitable algorithm parameter values should be initialized.
Sampling technique	<ul style="list-style-type: none"> - Perform easily and simply. - Efficient for both polynomial and rational Lyapunov function. 	<ul style="list-style-type: none"> - May diverge - Needs intensive numerical simulation to fix the solution. - Risk of stagnation in local optimum.
Analytical method	<ul style="list-style-type: none"> - A step by step recursive algorithm is well defined analytically. - No dummy solution can be found. 	<ul style="list-style-type: none"> - May diverge - Delicate algebraic and analytical developments are indispensable. - Risk of stagnation in local optimum - Polynomial function approximation is needed.

An advanced metaheuristic strategy pointing to maximizing the DA of stable equilibria using Lyapunov theory is established in this work. The estimated region is fully bounded by the domain of the LF negative definiteness and its time derivative. The CKH optimizer is designed such that the requirements on the sign of the LF and the tangency constraint between the level sets are satisfied. As a result, numerous potential local solutions have been avoided by such constraints.

Unlike existing techniques, the developed method can be applied for both polynomial and rational LF. Indeed, it allows the development of a chaotic operative optimization algorithm which ensures the convergence to an enlarged DA in a significantly limited running time. Despite the performance, offering a larger accurate DA at high speed converging time, CKH is valuable for real-time trajectory tracking control problems.

To highlight the performance of the CKH technique a comparative analysis is performed with a swarm-intelligence based approach. The study compares the CKH with the CPSO technique while considering the performance criteria as the DA radius, the running time (ms) and the capability to provide an accurate description of the tangency point. The result of this study is summarized in Table 3.

Table 3. Estimated DA features for CKH and CPSO methods.

Example	Systems Dynamic	Lyapunov Function	Domain Radius		Running Time (ms)		Tangency Point	
			CPSO	CKH	CPSO	CKH	CPSO	CKH
E2 [28,29]	$\begin{cases} \dot{x}_1 = -x_2 \\ \dot{x}_2 = x_1 - x_2 + x_1^2 x_2 \end{cases}$	$V(x) = 1.5x_1^2 - 0.5x_1x_2 + x_2^2$	2.3045	2.3045	1.821	0.542	$\begin{cases} x_1 = -0.8575 \\ x_2 = 0.7483 \end{cases}$	$\begin{cases} x_1 = 0.852 \\ x_2 = -0.755 \end{cases}$

It appears in this table that both methods are highly comparable in term of provided performance. However, the CKH showed a superiority in the operating consumed time. This concludes that the CKH is an efficient, fast and accurate optimization technique.

6. Conclusions

The CKH algorithm provided the largest solution for the DA optimization. It was implemented in this work to compute the QLF, and allows expanding of the DA for a large class of nonlinear systems. The main objective of computing the largest sublevel set of the DA was perfectly achieved. The CKH-based iterative technique was developed to calculate DA lower bound by exploiting CKH capacity and implementing a novel enhanced KH. The synthesized design establishes chaotic maps for global optimization analysis and encodes the systems' states that will be determined as particle positions.

The observations and conclusions of this study are summarized below:

- The CKH technique suggested in the present study is straightforward, flexible, and easily understood. There are no challenges concerning the timing of parameter tuning; hence, it may be implemented for complex computational optimizations.
- The outcomes of the CKH algorithm, when benchmarked against the results of other popular optimization techniques, indicate the flexibility of the suggested approach.
- The proposed CKH technique requires less CPU time compared to other techniques because of the integration of chaos maps that provide chaotic maps benefits.

Hence, the benefits associated with the proposed technique used for estimating the DA showed that the concurrent use of KH with chaotic maps leads to rapid convergence and provides superior performance and better results. The chaotic maps produce a search space different from that produced by the KH process.

Motivated by the efficient results attained for the studied benchmark examples, the developed approach can be applied for real time tracking control problems with delicate time budget constraints. Theoretical concerns related to rational LF are a challenging aspect and a key perspective promoting this work.

Author Contributions: Conceptualization, F.H., H.J. and M.A.; methodology, F.H. and H.J.; software, M.A. and F.H.; validation, M.O. and D.P.; formal analysis, F.H. and H.J.; investigation, H.J. and M.A.; resources, R.A. and M.O.; data curation, D.P. and R.A.; writing—original draft preparation, F.H. and M.A.; writing—review and editing, H.J.; visualization, D.P.; supervision, M.O.; project administration F.H., H.J. and R.A.; funding acquisition, H.J. and M.O. All authors have read and agreed to the published version of the manuscript.

Funding: This project was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, under grant No. (D-74-305-1442). The authors, therefore, gratefully acknowledge DSR technical and financial support.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Theoretical Background, Fundamentals and Notations

Equation (A1) describes a dynamical nonlinear affine autonomous model:

$$\dot{x} = f(x), \quad x \in R^n; \quad x_0 = x(t_0) \tag{A1}$$

Consider $x = \bar{x}$ as an asymptotically stable equilibrium position corresponding to (A1).

Preserving generality, we assume that \bar{x} corresponds to the origin of the state space $\bar{x} = 0$. Now, let $x(t, x_0)$ define the trajectory of a system with an initial state x_0 at time t_0 .

Definition A1 (Equilibrium Point). *A point \bar{x} corresponds to a system equilibrium if*

$$f(\bar{x}) = 0 \tag{A2}$$

Definition A2 (Asymptotic Stability). *Considering the system (A1), the equilibrium $\bar{x} = 0$ is asymptotically stable if a constant $\lambda > 0$ can be identified to satisfy the below-mentioned condition: when $\|x\| > \lambda$,*

$$\lim_{t \rightarrow \infty} x(t, x_0) = 0 \tag{A3}$$

Definition A3 (Khalil [30]). *The domain or region of attraction (DA) specific to the origin of (A1) is specified as:*

$$S = \left\{ x_0 \in R^n : \lim_{t \rightarrow \infty} x(t, x_0) \rightarrow 0 \right\} \tag{A4}$$

Lyapunov stability concepts offer a way to evaluate equilibrium stability using LF.

Definition A4 (La Salle and Lefschetz [29], Lyapunov Function (LF), Hahn [39]). *Consider $V(x)$ as a real function that is continuously differentiable and specific on the domain $D \subseteq R^n$ having the origin. Then, $V(x)$ is a LF for the system (A1) if the following requirements are met:*

- $V(x)$ has a positive and definite value in D ,
- Considering the trajectories of system (A1), the time derivative of $V(x)$

$$\dot{V}(x) = [\nabla V(x)]^T \cdot f(x) \tag{A5}$$

is semidefinite and negative in D .

Definition A5 (Negative and positive definite functions). *Consider a continuously differentiable real function $\phi(x)$ specified using domain $D \subseteq R^n$ and containing the coordinate $x = 0$. The function is positive definite if the specified constraints are met:*

$$\phi(0) = 0 \tag{A6}$$

$$\phi(x) > 0 \forall x \in \{D/0\} \tag{A7}$$

If $-\phi(x)$ is positive definite, then $\phi(x)$ is negative definite.

Theorem A1 (Guaranteed estimation of the domain of attraction, La Salle and Lefschetz [29]). *Consider $V(x)$ as a continuous real function that is differentiable in its domain. Assume $V(x)$ to be a LF corresponding to system (A1) equilibrium $x = 0$. Consider that $\frac{dV(x)}{dt}$ is negative definite in the area specified by the following expression:*

$$\Omega(0) = \{x : V(x) = c, c > 0\} \tag{A8}$$

Under these conditions, the origin is asymptotically stable; all trajectories defined within $\Omega(0)$ approach zero as time approaches infinity.

References

1. Díaz, D.; Valledor, P.; Ena, B.; Iglesias, M.; Menéndez, C. Improved Method for Parallelization of Evolutionary Metaheuristics. *Mathematics* **2020**, *8*, 1476. [[CrossRef](#)]
2. Stefanoiu, D.; Borne, P.; Popescu, D.; Filip, F.G.; El Kamel, A. Metaheuristics—Local Methods. *Optim. Eng. Sci. Approx. Metaheuristic Methods* **2014**, 1–52. [[CrossRef](#)]
3. Kaveh, A.; Ghazaan, M.I. *Meta-Heuristic Algorithms for Optimal Design of Real-Size Structures*; Springer Science and Business Media LLC: Cham, Switzerland, 2018.
4. Hamidi, F.; Jerbi, H.; Aggoune, W.; Djemai, M.; Abdkrim, M.N. Enlarging Region of Attraction Via LMI-Based Approach and Genetic Algorithm. In Proceedings of the 2011 International Conference on Communications, Computing and Control Applications (CCCA), Hammamet, Tunisia, 3–5 March 2011; pp. 1–6.
5. Syafrudin, M.; Alfian, G.; Fitriyani, N.L.; Anshari, M.; Hadibarata, T.; Fatwanto, A.; Rhee, J. A Self-Care Prediction Model for Children with Disability Based on Genetic Algorithm and Extreme Gradient Boosting. *Mathematics* **2020**, *8*, 1590. [[CrossRef](#)]
6. Ma, Z.; Yuan, X.; Han, S.; Sun, D. Improved Chaotic Particle Swarm Optimization Algorithm with More Symmetric Distribution for Numerical Function Optimization. *Symmetry* **2019**, *11*, 876. [[CrossRef](#)]
7. Khan, S.; Kamran, M.; Rehman, O.U.; Liu, L.; Yang, S. A modified PSO algorithm with dynamic parameters for solving complex engineering design problem. *Int. J. Comput. Math.* **2017**, *95*, 2308–2329. [[CrossRef](#)]
8. Jaber, A.S.; Abdulbari, H.A.; Shalash, N.A.; Abdalla, A.N. Garra Rufa-inspired optimization technique. *Int. J. Intell. Syst.* **2020**, *35*, 1831–1856. [[CrossRef](#)]
9. Bansal, J.C.; Singh, P.K.; Pal, N.R. (Eds.) *Evolutionary and Swarm Intelligence Algorithms*; Springer: Cham, Switzerland, 2019.
10. Tilahun, S.L.; Ngotchouye, J.M.T. Firefly algorithm for discrete optimization problems: A survey. *KSCE J. Civ. Eng.* **2017**, *21*, 535–545. [[CrossRef](#)]
11. Deng, W.; Xu, J.; Zhao, H. An Improved Ant Colony Optimization Algorithm Based on Hybrid Strategies for Scheduling Problem. *IEEE Access* **2019**, *7*, 20281–20292. [[CrossRef](#)]
12. Wu, G.; Shen, X.; Li, H.; Chen, H.; Lin, A.; Suganthan, P. Ensemble of differential evolution variants. *Inf. Sci.* **2018**, *423*, 172–186. [[CrossRef](#)]
13. Pradeepmon, T.G.; Panicker, V.V.; Sridharan, R. A heuristic algorithm enhanced with probability-based incremental learning and local search for dynamic facility layout problems. *Int. J. Appl. Decis. Sci.* **2018**, *4*, 352–389. [[CrossRef](#)]
14. Prayogo, D.; Cheng, M.Y.; Wu, Y.W.; Herdany, A.A.; Prayogo, H. Differential Big Bang-Big Crunch algorithm for construction-engineering design optimization. *Autom. Constr.* **2018**, *85*, 290–304. [[CrossRef](#)]
15. Mirjalili, S. Biogeography-Based Optimization. In *Evolutionary Algorithms and Neural Networks*; Springer: Cham, Switzerland, 2019; pp. 57–72.
16. Ouyang, H.B.; Gao, L.Q.; Li, S.; Kong, X.Y.; Wang, Q.; Zou, D.X. Improved harmony search algorithm: LHS. *Appl. Soft Comput.* **2017**, *53*, 133–167. [[CrossRef](#)]
17. Mareli, M.; Twala, B. An adaptive Cuckoo search algorithm for optimization. *Appl. Comput. Inform.* **2018**, *2*, 107–115. [[CrossRef](#)]
18. Li, X.; Zhang, J.; Yin, M. Animal migration optimization: An optimization algorithm inspired by animal migration behavior. *Neural Comput. Appl.* **2014**, *7*, 1867–1877. [[CrossRef](#)]
19. Singh, G.P.; Singh, A. Comparative study of krill herd, firefly and cuckoo search algorithms for unimodal and multimodal optimization. *Int. J. Intell. Syst. Appl. Eng.* **2014**, *3*, 26–37. [[CrossRef](#)]
20. Cui, Z.; Li, F.; Zhang, W. Bat algorithm with principal component analysis. *Int. J. Mach. Learn. Cybern.* **2019**, *3*, 603–622. [[CrossRef](#)]
21. Mishra, M.; Gunturi, V.R.; Maity, D. Teaching-learning-based optimization algorithm and its application in capturing critical slip surface in slope stability analysis. *Soft Comput.* **2020**, *4*, 2969–2982. [[CrossRef](#)]
22. Talatahari, S.; Motamedi, P.; Azar, B.F.; Azizi, M. Tribe-charged system search for parameter configuration of nonlinear systems with large search domains. *Eng. Optim.* **2021**, *53*, 18–31. [[CrossRef](#)]
23. Asteris, P.G.; Nozhati, S.; Nikoo, M.; Cavaleri, L.; Nikoo, M. Krill herd algorithm-based neural network in structural seismic reliability evaluation. *Mech. Adv. Mater. Struct.* **2018**, *26*, 1146–1153. [[CrossRef](#)]
24. Wang, G.-G.; Gandomi, A.H.; Alavi, A.H.; Gong, D. A comprehensive review of krill herd algorithm: Variants, hybrids and applications. *Artif. Intell. Rev.* **2019**, *51*, 119–148. [[CrossRef](#)]
25. Abualigah, L.; Khader, A.T.; Hanandeh, E.S. A combination of objective functions and hybrid Krill herd algorithm for text document clustering analysis. *Eng. Appl. Artif. Intell.* **2018**, *73*, 111–125. [[CrossRef](#)]
26. Abualigah, L.M.; Khader, A.T.; Hanandeh, E.S.; Gandomi, A. A novel hybridization strategy for krill herd algorithm applied to clustering techniques. *Appl. Soft Comput.* **2017**, *60*, 423–435. [[CrossRef](#)]
27. Wang, G.-G.; Guo, L.; Gandomi, A.; Hao, G.-S.; Wang, H. Chaotic Krill Herd algorithm. *Inf. Sci.* **2014**, *274*, 17–34. [[CrossRef](#)]
28. Najafi, E.; Babuška, R.; Lopes, G.A. A fast sampling method for estimating the domain of attraction. *Nonlinear Dyn.* **2016**, *2*, 823–834. [[CrossRef](#)]
29. Hachicho, O. A novel LMI-based optimization algorithm for the guaranteed estimation of the domain of attraction using rational Lyapunov functions. *J. Frankl. Inst.* **2007**, *344*, 535–552. [[CrossRef](#)]
30. Khalil, H.K. *Nonlinear Systems*, 3rd ed.; Prentice Hall: Upper Saddle River, NJ, USA, 2002.
31. Chesi, G. Rational Lyapunov functions for estimating and controlling the robust domain of attraction. *Automatica* **2013**, *49*, 1051–1057. [[CrossRef](#)]

32. Jerbi, H.M.; Hamidi, F.; Ben Aoun, S.; Olteanu, S.C.; Popescu, D. Lyapunov-based Methods for Maximizing the Domain of Attraction. *Int. J. Comput. Commun. Control.* **2020**, *15*, 5. [[CrossRef](#)]
33. Chesi, G. Computing Output Feedback Controllers to Enlarge the Domain of Attraction in Polynomial Systems. *IEEE Trans. Autom. Control.* **2004**, *49*, 1846–1850. [[CrossRef](#)]
34. Ran, M.; Wang, Q.; Dong, C.; Ni, M. Multistage anti-windup design for linear systems with saturation nonlinearity: Enlargement of the domain of attraction. *Nonlinear Dyn.* **2015**, *3*, 1543–1555. [[CrossRef](#)]
35. Hamidi, F.; Jerbi, H.; Aggoune, W.; Djemaï, M.; Abdelkrim, M.N. Enlarging the Domain of Attraction in Nonlinear Polynomial Systems. *Int. J. Comput. Commun. Control.* **2013**, *8*, 538. [[CrossRef](#)]
36. Ermolin, V.S.; Vlasova, T.V. Identification of the Domain of Attraction. In Proceedings of the 2015 International Conference “Stability and Control Processes” in Memory of V.I. Zubov (SCP), Saint Petersburg, Russia, 5–9 October 2015; pp. 9–12.
37. Doban, A.I.; Lazar, M. Computation of Lyapunov functions for nonlinear differential equations via a Massera-type construction. *IEEE Trans. Autom. Control.* **2017**, *5*, 1259–1272. [[CrossRef](#)]
38. Chesi, G. Domain of Attraction: Estimates for Non-Polynomial Systems Via LMIs. In Proceedings of the 16th IFAC World Congress, Prague, Czech Republic, 3–8 July 2005.
39. Hamidi, F.; Jerbi, H. On the Estimation of a Maximal Lyapunov Function and Domain of Attraction Determination Via a Genetic Algorithm. In Proceedings of the 2009 6th International Multi-Conference on Systems, Signals and Devices, Djerba, Tunisia, 23–26 March 2009; pp. 1–6.
40. Zheng, X.; She, Z.; Lu, J.; Li, M. Computing multiple Lyapunov-like functions for inner estimates of domains of attraction of switched hybrid systems. *Int. J. Robust Nonlinear Control.* **2018**, *17*, 5191–5212. [[CrossRef](#)]
41. Hamidi, F.; Jerbi, H.; Olteanu, S.C.; Popescu, D. An Enhanced Stabilizing Strategy for Switched Nonlinear Systems. *Stud. Inform. Control.* **2019**, *28*, 391–400. [[CrossRef](#)]
42. Chesi, G.; Colaneri, P. Homogeneous rational Lyapunov functions for performance analysis of switched systems with arbitrary switching and dwell time constraints. *IEEE Trans. Autom. Control.* **2017**, *10*, 5124–5137. [[CrossRef](#)]
43. Jerbi, H.; Braiek, N.B.; Bacha, A.B.B. A method of estimating the domain of attraction for nonlinear discrete-time systems. *Arab. J. Sci. Eng.* **2014**, *5*, 3841–3849. [[CrossRef](#)]
44. Chesi, G.; Garulli, A.; Tesi, A.; Vicino, A. LMI-based computation of optimal quadratic Lyapunov functions for odd polynomial systems. *Int. J. Robust Nonlinear Control.* **2004**, *15*, 35–49. [[CrossRef](#)]
45. Pitarch, J.L.; Sala, A.; Arino, C.V. Closed-Form Estimates of the Domain of Attraction for Nonlinear Systems via Fuzzy-Polynomial Models. *IEEE Trans. Cybern.* **2013**, *44*, 526–538. [[CrossRef](#)] [[PubMed](#)]
46. Luk, C.K.; Chesi, G. On the estimation of the domain of attraction for discrete-time switched and hybrid nonlinear systems. *Int. J. Syst. Sci.* **2015**, *15*, 2781–2787. [[CrossRef](#)]
47. Hamidi, F.; Aloui, M.; Jerbi, H.; Kchaou, M.; Abbassi, R.; Popescu, D.; Ben Aoun, S.; Dimon, C. Chaotic Particle Swarm Optimisation for Enlarging the Domain of Attraction of Polynomial Nonlinear Systems. *Electronics* **2020**, *9*, 1704. [[CrossRef](#)]
48. Zečević, A.I.; Šiljak, D.D. Estimating the region of attraction for large-scale systems with uncertainties. *Automatica* **2010**, *46*, 445–451. [[CrossRef](#)]
49. Topcu, U.; Packard, A.K.; Seiler, P.; Balas, G.J. Robust Region-of-Attraction Estimation. *IEEE Trans. Autom. Control.* **2009**, *55*, 137–142. [[CrossRef](#)]
50. Swiatlak, R.; Tibken, B.; Paradowski, T.; Dehnert, R. An Interval Arithmetic Approach for the Estimation of the Robust Domain of Attraction for Nonlinear Autonomous Systems with Nonlinear Uncertainties. In Proceedings of the 2015 American Control Conference (ACC), Chicago, IL, USA, 1–3 July 2015; pp. 2679–2684.
51. Nersesov, S.G.; Ashrafiuon, H.; Ghorbanian, P. On estimation of the domain of attraction for sliding mode control of underactuated nonlinear systems. *Int. J. Robust Nonlinear Control.* **2014**, *5*, 811–824. [[CrossRef](#)]
52. Majumdar, A.; Vasudevan, R.; Tobenkin, M.; Tedrake, R. Convex Optimization of Nonlinear Feedback Controllers via Occupation Measures. *Int. J. Robot. Res.* **2014**, *9*, 1209–1230. [[CrossRef](#)]
53. Jerbi, H. Estimations of the Domains of Attraction for Classes of Nonlinear Continuous Polynomial Systems. *Arab. J. Sci. Eng.* **2017**, *42*, 2829–2837. [[CrossRef](#)]
54. Jemai, W.J.; Jerbi, H.; Abdelkrim, M.N. Nonlinear state feedback design for continuous polynomial systems. *Int. J. Control. Autom. Syst.* **2011**, *9*, 566–573. [[CrossRef](#)]
55. Baier, R.; Gerdtts, M. A Computational Method for Non-Convex Reachable Sets Using Optimal Control. In Proceedings of the 2009 European Control Conference (ECC), Budapest, Hungary, 23–26 August 2009; pp. 97–102.
56. Bacanin, N.; Bezdan, T.; Tuba, E.; Strumberger, I.; Tuba, M. Monarch Butterfly Optimization Based Convolutional Neural Network Design. *Mathematics* **2020**, *8*, 936. [[CrossRef](#)]
57. Baghban, A.; Jalali, A.; Shafiee, M.; Ahmadi, M.H.; Chau, K.-W. Developing an ANFIS-based swarm concept model for estimating the relative viscosity of nanofluids. *Eng. Appl. Comput. Fluid Mech.* **2019**, *13*, 26–39. [[CrossRef](#)]
58. Matallana, L.G.; Blanco, A.M.; Bandoni, J.A. Estimation of domains of attraction: A global optimization approach. *Math. Comput. Model.* **2010**, *52*, 574–585. [[CrossRef](#)]