A Novel Improved Sea-Horse Optimizer for Tuning Parameter Power System Stabilizer

Widi Aribowo *
Department of Electrical Engineering, Universitas Negeri Surabaya, Surabaya, Indonesia
Email: widiarıbowo@unesa.ac.id
*Corresponding Author

Abstract— Power system stabilizer (PSS) is applied to dampen system oscillations so that the frequency does not deviate beyond tolerance. PSS parameter tuning is increasingly difficult when dealing with complex and nonlinear systems. This paper presents a novel hybrid algorithm developed from incorporating chaotic maps into the sea-horse optimizer. The algorithm developed is called the chaotic sea-horse optimizer (CSHO). The proposed method is adopted from the metaheuristic method, namely the sea-horse optimizer (SHO). The SHO is a method that duplicates the life of a sea-horse in the ocean when it moves, looks for prey and breeds. In this paper, The CSHO method is used to tune the power system stabilizer parameters on a single machine system. The proposed method validates the benchmark function and performance on a single machine system against transient response. Several metaheuristic methods are used as a comparison to determine the effectiveness and efficiency of the proposed method. From the research, it was found that the application of the logistics Tent map from the chaotic map showed optimal performance. In addition, the application of the PSS shows effective and efficient performance in reducing overshoot in transient conditions.

Keywords— Smlb; Chaotic sea-horse optimizer; Metaheuristic; Power system stabilizer; Power system.

I. INTRODUCTION

Rapid technological developments indirectly affect the stability and efficiency of the power system [1]–[4]. Low frequency oscillations are a big challenge in power systems [5]–[10]. This affects the toughness of the system as a single machine or interconnection if it cannot be responded to appropriately [11]–[15]. Inappropriate response will affect the electricity supply which results in economic losses. Tremendous negative effect on system stability caused by low frequency oscillations so that maximum effort is required. The stability of the power system is a key word that must always be maintained [16]–[21].

Power System Stabilizers (PSS) are a well-known and effective approach to dealing with low frequency oscillations [22]–[27]. PSS is applied to maintain the stability of the power system either as a single machine or interconnection [28]–[33]. PSS is designed with the nature of the system in mind. The non-linear characteristics of the power system and consistent fluctuations over a wide range can make conventional PSS insufficient to achieve optimal performance. In addition, conventional PSS has approaches such as self-tuning regulators and feedback linearization. However, this approach presents drawbacks such as intensive computing and long computer processing times [34]–[36]. To overcome these problems and produce more efficient and optimal solutions. Several algorithmic approaches have been presented by researchers as alternative methods.

The development of computing technology indirectly encourages the development of several new algorithms. There has been a significant increase from the discovery of new metaheuristic methods in recent years to get better optimizations for non-linear and complex problems. The application of traditional deterministic optimization methods often encounters several problems such as balance in exploitation-exploration, deadlock with local optimization and cannot be separated from derivatives. Metaheuristic algorithm has the characteristics of a simple, and optimal approach. In addition, metaheuristic algorithms are also durable and self-organized. Some of the latest metaheuristic algorithms in recent years such As FOX [37], Giant Trevally Optimizer [38], Dung beetle optimizer [39], Ebola optimization search algorithm [40], Dwarf mongoose optimization algorithm [41], Snake Optimizer [42], Arithmetic Optimization Algorithm [43], Archimedes Optimization Algorithm [44], Remora Optimization Algorithm [45], African Vultures Optimization Algorithm [46], Horse Herd Optimization Algorithm [47], Battle Royale Optimization Algorithm [48], Chimp Optimization Algorithm [49], Pelican Optimization Algorithm [50], Prairie Dog Optimization Algorithm [51], and Group Search Optimizer [52]. The average output of the applied method produces the optimal approach for a particular solution.

The application of metaheuristic methods in adjusting the power system stabilizer has provided promising performance. Several metaheuristic algorithms have been applied to power system stabilizers in recent years, such as Atomic Search Optimization [53], Ant Colony Optimization Algorithm [54], [55], Crow Search Algorithm [56], Tunicate Swarm Algorithm [57], Harris Hawk Optimizer [58], [59], Moth Search Algorithm [60], [61], Mayfly Optimization Algorithm [62], Sine-Cosine Algorithm [63], Rat Swarm Optimization [64], Whale Optimization Algorithm [65], [66] and Particle Swarm Optimization [67]–[70].

Several researchers have also presented a combination of metaheuristic algorithms for tuning PSS parameters, such as Gude et al demonstrated a combination of butterfly optimization algorithm and particle swarm optimization [71]. Kalegowda et al presented a combination of Particle swarm optimization and Taguchi algorithm [72]. Devarapalli et al presented a combination of gray wolf optimization, sine cosine algorithm and cuckoo search [73]. Penchalaiah et al...
presented a combination of Harris Hawks Optimization Algorithm and the Tabu Search Algorithm [74]. Although several metaheuristic algorithms have been demonstrated for PSS tuning, there is still a lot of room to be explored to get an optimal PSS performance. Therefore, this study presents a PSS parameter tuning approach with the latest metaheuristic algorithm called Sea-Horse Optimizer (SHO) which is improved by adding a chaotic algorithm [75]. This hybrid method is called the Chaotic Sea-Horse Optimizer (CSHO) method which aims to improve the performance of the SHO algorithm at the center of the balance between exploration and exploitation. The proposed method is an integration of chaotic and SHO methods. The contribution of this research can be briefly described as follows:

1. A hybrid algorithm is presented, namely Chaotic Sea-Horse Optimization (CSHO) which has a new balance between exploration and exploitation phases.

2. The proposed CSHO is applied to tune the power system stabilizer parameters.

This article consists of several sessions, namely: Part 2 explains the Sea-Horse Optimizer method, the novel Chaotic Sea-Horse Optimizer and power system stabilizers. Session 3 contains the design of the proposed control. The simulation results and discussion are presented in section 4. The last section contains conclusions and future works.

II. METHODS

A. Sea-Horse Optimizer (SHO)

The sea-horse optimizer (SHO) is inspired by the life of seahorses while searching for prey, movement and breeding in the sea. The concept of exploration and exploitation that characterizes the metaheuristic method in the SHO algorithm is designed to adopt the social behavior of movement and search for seahorse prey. The last phase of breeding is executed when the two components have ended. The SHO method can be modeled in detail as follows:

\[
Sh = \begin{bmatrix}
x_{1,i} & \cdots & x_{1,Dim-1} & x_{1,Dim} \\
x_{2,i} & \cdots & x_{2,Dim-1} & x_{2,Dim} \\
\vdots & \ddots & \vdots & \vdots \\
x_{N,i} & \cdots & x_{N,Dim-1} & x_{N,Dim} \\
\end{bmatrix}
\]  

(1)

\[
Sh_{ij} = \text{Rand} \times (UB_j - LB_j) + LB_j
\]  

(2)

\[
Sh_{\text{elite}} = \text{argmin}(f(X_i))
\]  

(3)

Where Dim is the dimension of the variable and N is size of population. The upper and lower limits are symbolized by UB and LB which are random results from each solution. Random value with a range from 0 to 1 is denoted by Rand. Sh_{\text{elite}} is a symbol of individuals who have a minimum level of fitness represented as elite individuals. SHO adopts the life of seahorses in the form of movement, looking for prey and breeding.

Seahorse Movement Behavior

The normal distribution becomes a reference in the movement pattern of seahorses, balance of exploration and exploitation using two case studies with a boundary point of 0.

Case 1: the agent moves towards the \( X_{\text{elite}} \) in a spiral motion and changes the rotation angle constantly to widen the local solution region. Case 1 can be formulated mathematically as follows:

\[
X_n(t + 1) = X_i(t) + \text{Levy}(\lambda)((X_{\text{elite}}(t) - X_i(t)) \times x \times x \times x \times z + X_{\text{elite}}(t))
\]  

(4)

\[
\theta = \text{Rand} \times 2\pi
\]  

(5)

\[
x = \cos(\theta)
\]  

(6)

\[
y = \rho \times \sin(\theta)
\]  

(7)

\[
z = \rho \times \theta
\]  

(8)

\[
\rho = \mu \times e^{\theta v}
\]  

(9)

\[
\text{Levy}(\lambda) = s \times \frac{w \times \sigma}{|K|^{\frac{\lambda}{2}}}
\]  

(10)

\[
\sigma = \left( \frac{\Gamma(1 + \lambda) \sin \left( \frac{\pi \lambda}{2} \right)}{\Gamma \left( \frac{1 + \lambda}{2} \right) \times \lambda \times 2^{\frac{\lambda - 2}{2}}} \right)
\]  

(11)

Where the length of the rod specified by the logarithmic spiral constant \( \mu \) (default=0.05) and \( v \) (default=0.05) is denoted by \( \rho \). \( x \) is a random number \([0, 2]\), \( k \) and \( w \) are random numbers \([0, 1]\), \( s \) is a fixed constant of 0.01.

Case 2: with ocean waves, the seahorse performs a brownian motion that mimics the motion length of another seahorse in an attempt to get a better traverse. This can be formulated as follows:

\[
X_n(t + 1) = X_i(t) + \text{Rand} \times l \times \beta_i \times (X_i(t) - X_{\text{elite}}(t))
\]  

(12)

\[
\beta_i = \frac{1}{\sqrt{2\pi}} \exp \left(-\frac{x^2}{2}\right)
\]  

(13)

\[
X_n(t + 1) = \begin{cases} 
X_i(t) + \text{Levy}(\lambda)((X_{\text{elite}}(t) - X_i(t)) \times x \times y \times z + X_{\text{elite}}(t)) & \text{if } r_1 > 0 \\
X_i(t) + \text{Rand} \times l \times \beta_i \times (X_i(t) - X_{\text{elite}}(t)) & \text{if } r_1 \leq 0 
\end{cases}
\]  

(14)

where \( \beta_i \) is the random walk coefficient of the brownian motion. \( l \) is a constant value (default=0.5). \( r_1 \) is symbolized as a random value

Seahorse Foraging Behavior

When seahorses look for food there are 2 possible outcomes, namely success and failure. The condition is successfully set with a value of \( r_2 > 0 \). This condition when the seahorse moves faster than the prey. On the other hand, a failure condition occurs when the response is different. The conditions of failure and success of seahorses when looking for food can be formulated as follows:

\[
X_n(t + 1) = \begin{cases} 
\alpha \times ((X_{\text{elite}}(t) - \text{Rand} \times X_{\text{new}}(t)) + (1 - \alpha) \times X_{\text{elite}}(t)) & \text{if } r_2 > 0 \\
(1 - \alpha) \times ((X_{\text{new}}(t) - \text{Rand} \times X_{\text{elite}}(t)) + (\alpha) \times X_{\text{new}}(t)) & \text{if } r_2 \leq 0 
\end{cases}
\]  

(15)
where $X_{\text{new}}$ is the new position of the seahorse. $r_2$ is a random number $[0,1]$, $T$ is the maximum iteration.

**Seahorse Breeding Behavior**

At the time of breeding, seahorses are divided into 2 sex groups, namely male and female with the same amount of composition, namely 50%.

$\text{Father} = X_{\text{sort}} \left(1: \frac{\text{pop}}{2}\right)$ \hspace{1cm} (17)

$\text{Mother} = X_{\text{sort}} \left(\frac{\text{pop}}{2} + 1 : \text{pop}\right)$ \hspace{1cm} (18)

Sorted $X_{\text{sort}}$ values return the result in ascending order. $\text{Father}$ and $\text{Mother}$ were chosen randomly. In the SHO algorithm, each pair produces one child.

$X_i = (1 - r_3)X_{\text{mother}} + r_3X_{\text{father}}$ \hspace{1cm} (19)

Where $r_3$ is a random number $[0,1]$.

**B. The Novel Chaotic Sea-Horse Optimizer (CSHO)**

Several articles have used several types of chaotic maps for the purpose of algorithm optimization. Chaotic maps are dynamic in character and statistics built on randomness [76], [77]. Future or future behavior is affected by parameter changes. So that small changes to the parameters produce different outputs. In this article, the logistic type chaotic map is used to replace the $\text{rand}$ in Equation (5). The mathematical equation of the logistic type chaotic map is as follows:

$y_{log(i+1)} = a \times y_{log(i)} \left(1 - y_{log(i)}\right)$ \hspace{1cm} (20)

Where $a$ is 4. So, the chaotic map variable with range $[0,1]$ is obtained. So, Equation (5) turns into Equation (21) as follows:

$\theta = y_{\log} \times 2\pi$ \hspace{1cm} (21)

Pseudo code from CSHO can be seen in Algorithm 1.

### Algorithm 1 Pseudo Code of Chaotic Sea-Horse Optimizer

**Input:** population size pop, maximum iteration T and variable dimension Dim

**Output:** Optimal search agent Xbest

1: procedure CSHO
2: Initialize search agent Xi
3: Calculate the fitness value of each search agent
4: Determine the best search agent Xe
5: /* Movement behavior */
6: While ($t < \text{Max} \_ \text{iteration}$) do
7: $u \leftarrow 0.05$
8: $v \leftarrow 0.05$
9: $l \leftarrow 0.05$
10: $y_{log(i+1)} \leftarrow a \times y_{log(i)} \left(1 - y_{log(i)}\right)$ using Eq. (20)
11: $\theta \leftarrow y_{log} \times 2\pi$ using Eq. (21)
12: $x \leftarrow \cos(\theta)$ using Eq. (6)
13: $y \leftarrow \rho \times \sin(\theta)$ using Eq. (7)
14: $z \leftarrow \rho \times \theta$ by using Eq. (8)
15: $\rho \leftarrow \mu \times e^{8\pi}$ by using Eq. (9)
16: else if do
17: update positions of the search agent by using Eq. (12)
18: end if
*/ Foraging behavior */
19: update positions of the search agent by using Eq. (15)
20: Calculate the fitness value of each search agent
*/ Foraging behavior */
21: Select fathers and mother by using Eq. (17) and Eq. (18)
22: Breed Offspring by using Eq. (19)
23: Calculate the fitness value of each offspring
24: Select the next iteration population from the offspring and parent ranked top pop in fitness values
25: Update elite ($X_{\text{elite}}$) position
26: $t \leftarrow t + T$
27: end while
28: end procedure

**C. Power System Stabilizer**

The main function of PSS is to produce a suitable damping torque on the engine rotor. This aims to obtain compensation from the phase lag between the excitation input and the electric torque [78]–[80]. So that the PSS output is proportional to the rotor speed. The widely used conventional lead-lag PSS structure can be seen in Fig. 1.

![Fig. 1. PSS lead-lag type [81]](image)

The optimal value of the PSS lead-lag parameter was optimized using CSHO. This is to improve the closed-loop response of the system to the terms of the transient response criteria. Fig. 2 illustrates the block diagram of the proposed CSHO-PSS approach.

**III. DESIGN OF CONTROLLERS**

The optimal value of the PSS lead-lag parameter was optimized using CSHO. This is to improve the closed-loop response of the system to the terms of the transient response criteria. Fig. 2 illustrates the block diagram of the proposed CSHO-PSS approach. Some guidelines for getting PSS parameters with CSHO, namely:

- Step one: start by creating a single machine system
- Step two: design and build the CSHO algorithm
- Step three: Perform CSHO integration with single machine
- Step four: the system is run according to the desired constraints to get the PSS parameters
- Step five: Get the PSS parameters
- Step six: done
Fig. 2. Block diagram of the proposed CSHO algorithm implementation on PSS Lead-Lag.

**TABLE I. UNIMODAL FUNCTION[82], [83]**

<table>
<thead>
<tr>
<th>Test Function</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_1(x) = \sum_{i=1}^{n} x_i^2$</td>
<td>$[-100,100]^n$</td>
</tr>
<tr>
<td>$F_2(x) = \sum_{i=1}^{n}</td>
<td>x_i</td>
</tr>
<tr>
<td>$F_3(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} (x_j)^2$</td>
<td>$[-100,100]^n$</td>
</tr>
<tr>
<td>$F_4(x) = \max {</td>
<td>x_i</td>
</tr>
<tr>
<td>$F_5(x) = \sum_{i=1}^{n} [100(x_{i+1} - x_i)^2 + (x_i - 1)^2]$</td>
<td>$[-30,30]^n$</td>
</tr>
<tr>
<td>$F_6(x) = \sum_{i=1}^{n} (</td>
<td>x_i + 0.5</td>
</tr>
<tr>
<td>$F_7(x) = \sum_{i=1}^{n} 0.5 x_i^4 + \text{random}(0.1)$</td>
<td>$[-1.28,1.28]^n$</td>
</tr>
</tbody>
</table>
\[
F_3(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{x_i})
\]

\[
F_4(x) = \sum_{i=1}^{n} [x_i^2 - 10 \cos(2\pi x_i) + 10]
\]

\[
F_{10}(x) = -20 \exp(-0.2 \sqrt{n} \sum_{i=1}^{n} x_i^2) - \exp(-0.2 \sum_{i=1}^{n} \cos(2\pi x_i)) + 20 + e
\]

\[
F_{11}(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 + \frac{1}{1} \sum_{i=1}^{n} \cos \left( \frac{x_i}{\sqrt{10}} \right) + 1
\]

\[
F_{12}(x) = \frac{\pi}{2} \left[ \sum_{i=1}^{n} \left( x_i - 1 \right) \sum_{i=1}^{n} \left( y_i - 1 \right) \right] + \sum_{i=1}^{n} \left( x_i - 1 \right) \sum_{i=1}^{n} \left( y_i - 1 \right)
\]

\[
F_{13}(x) = 0.1 \left[ \sin^2(3\pi x_1) + \sum_{i=1}^{n} (X_i - 1)^2 \left( 1 + \sin^2(3\pi x_1) \right) + (X_i - 1)^2 \left( 1 + \sin^2(2\pi x_n) \right) \right]
\]

\[
F_{16}(x) = \left( \frac{1}{500} + \sum_{j=1}^{25} \frac{1}{1 + \sum_{i=1}^{n} (x_i - a_{ij})^6} \right)^{-1}
\]

\[
F_{17}(x) = \sum_{i=1}^{n} \left( x_i - \frac{1}{4} \right)^2 + \frac{5}{4} \sum_{i=1}^{n} x_i^2 - 6 \sum_{i=1}^{n} x_i + 10 \sum_{i=1}^{n} \frac{1}{2} x_i^2 + 10 \sum_{i=1}^{n} \frac{1}{2} x_i^2
\]

\[
F_{18}(x) = \left[ 1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_2^2 - 14x_2 + 6x_1 x_2 + 3x_2^2) \right] \times \left[ 30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_2 + 48x_2 - 36x_1 x_2 + 27x_2^2) \right]
\]

\[
F_{19}(x) = \sum_{i=1}^{m} C_i \exp \left( -\sum_{i=1}^{n} a_{ii} (X_i - p_i)^2 \right)
\]

\[
F_{20}(x) = \sum_{i=1}^{m} C_i \exp \left( -\sum_{i=1}^{n} a_{ii} (X_i - p_i)^2 \right)
\]

\[
F_{21}(x) = -\sum_{i=1}^{n} \left( X - a_i \right) (X - a_i)^T + C_i \right]^{-1}
\]

\[
F_{22}(x) = -\sum_{i=1}^{n} \left( X - a_i \right) (X - a_i)^T + C_i \right]^{-1}
\]

\[
F_{23}(x) = -\sum_{i=1}^{n} \left[ X - a_i \right] (X - a_i)^T + C_i \right]^{-1}
\]
IV. SIMULATION RESULT AND DISCUSSION

A. Convergence Profile

The performance measurement of the CSHO algorithm uses the benchmark function. Details of the benchmark functions used in this article can be seen in Table 1 to Table 3. Unimodal functions ranging from F1 to F7 can be seen in Table 1. While multimodal functions can be seen from F8 to F13 can be seen in Table 2. Functions F14 to F23 are multimodal functions with fixed dimensions can be seen in Table 3.

The MATLAB/Simulink software via a laptop with an Intel I5-5200 2.19 GHz processor and 8 GB RAM is used for programming algorithm codes, simulating transient response, and robustness. The parameters of the CSHO algorithm in detail are listed in Table IV.

The proposed CSHO algorithm uses a population size of 50 while the maximum iteration limit is set at 30. This study uses the GOA, WOA and SHO algorithms as comparisons. The results of the comparison with the benchmark function from F1 to F23 are presented in Fig. 3.

<table>
<thead>
<tr>
<th>Parameter CSHO</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number Of Sea-Horse (population)</td>
<td>50</td>
</tr>
<tr>
<td>$u$</td>
<td>0.05</td>
</tr>
<tr>
<td>$v$</td>
<td>0.05</td>
</tr>
<tr>
<td>$l$</td>
<td>0.05</td>
</tr>
<tr>
<td>$a$ (chaotic map parameter)</td>
<td>4</td>
</tr>
<tr>
<td>Maximum Iteration</td>
<td>30</td>
</tr>
</tbody>
</table>

The MATLAB/Simulink software via a laptop with an Intel I5-5200 2.19 GHz processor and 8 GB RAM is used for programming algorithm codes, simulating transient response, and robustness. The parameters of the CSHO algorithm in detail are listed in Table IV.
Fig. 3. Convergence Curve with Benchmark Function.

From the simulation results, the CSHO algorithm shows a good indication of the performance of the balance between the global and local search phases for optimization problems. In addition, the CSHO algorithm is also compared with the original SHO algorithm. The average convergence value of CSHO is lower than SHO.

B. Comparison Transient Response with Various Algorithm

The source of frequency and voltage constant in both angle and magnitude is an infinite bus. The Heffron-Philips model is used as a mathematical analysis for small signal stability analysis. The CSHO algorithm is applied with the aim of solving the non-linear optimization problem in obtaining the PSS parameter set. The expected parameters are expected to reduce wave oscillations optimally. The results of applying the CSHO algorithm to lead-lag PSS were compared with conventional PSS models, GOA, WOA and original SHO in two case studies.

In case study 1, the system is given a load of 100%. PSS parameters that have been obtained using the CSHO
algorithm were tested with 100% load. In addition, PSS with the CSHO algorithm is compared with other approaches and the results can be seen in Fig. 4. Table V is the PSS parameters of each approach. The graph of case study 1 is presented in Fig. 4.

In case study 1, the CSHO algorithm applied to PSS (PSS-CSHO) is able to reduce the overshoot and undershoot of the speed and rotor angle optimally. Comparison with the conventional PSS method (PSS-Conv), it is found that the PSS-CSHO algorithm is able to reduce overshoot by 6.79% for rotor angle and 7.25% for speed. Meanwhile, PSS-CSHO was able to reduce undershoot by 75.86% for angle and 5.96% for speed. Details of case study 1 are detailed in Table VI.

Comparison with PSS-WOA on the overshoot of the rotor angle, it was found that the PSS-WOA overshoot value was 47% better. Compared to PSS-CSHO. On the other hand, PSS-CSHO has a better speed overshoot of 61.87% compared to PSS-WOA.

Table V. The Result of Parameter PSS

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Kpss</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSS-CSHO</td>
<td>-166.92626</td>
<td>266.7264</td>
<td>146.11179</td>
<td>1.7443</td>
<td>104.8790</td>
</tr>
<tr>
<td>PSS-SHO</td>
<td>23.7725</td>
<td>238.2281</td>
<td>-191.7066</td>
<td>2.433</td>
<td>24.1114</td>
</tr>
<tr>
<td>PSS-GOA</td>
<td>-64.7484</td>
<td>3.2133</td>
<td>100</td>
<td>-100</td>
<td>99.98</td>
</tr>
<tr>
<td>PSS-WOA</td>
<td>-38.5627</td>
<td>100</td>
<td>100</td>
<td>11.4185</td>
<td>100</td>
</tr>
</tbody>
</table>

Table VI. The result of Case Study 1

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Rotor Angle Output</th>
<th>Speed Output</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Overshoot</td>
<td>Undershoot</td>
</tr>
<tr>
<td>PSS-CSHO</td>
<td>0.0208</td>
<td>-0.3658</td>
</tr>
<tr>
<td>PSS-SHO</td>
<td>0.0391</td>
<td>-0.4957</td>
</tr>
<tr>
<td>PSS-GOA</td>
<td>0.308</td>
<td>-0.55</td>
</tr>
<tr>
<td>PSS-WOA</td>
<td>0.0111</td>
<td>-0.7395</td>
</tr>
<tr>
<td>PSS-Conv</td>
<td>0.0887</td>
<td>-1.1244</td>
</tr>
</tbody>
</table>

Fig. 4. Transient Response of Case Study 1 (a) Speed Response (b) Rotor Angle Response

Table VII. The Result of Case Study 2

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Rotor Angle Output</th>
<th>Speed Output</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Overshoot</td>
<td>Undershoot</td>
</tr>
<tr>
<td>PSS-CSHO</td>
<td>0.0104</td>
<td>-0.1829</td>
</tr>
<tr>
<td>PSS-SHO</td>
<td>0.0195</td>
<td>-0.2478</td>
</tr>
<tr>
<td>PSS-GOA</td>
<td>0.0154</td>
<td>-0.2750</td>
</tr>
<tr>
<td>PSS-WOA</td>
<td>0.0056</td>
<td>-0.3697</td>
</tr>
<tr>
<td>PSS-Conv</td>
<td>0.044</td>
<td>-0.5622</td>
</tr>
</tbody>
</table>
The load is reduced by 50% in the case study 2. The response from case study 2 can be seen in detail in Table VII and the comparison graph can be seen in Fig. 5. PSS-CSHO has the optimal ability to reduce overshoot and undershoot in case study 2. Comparison with the method PSS-Conv found that the undershoot and overshoot of the PSS-CSHO method is better by 3.36% and 37.93% on the rotor angle. While on speed, PSS-CSHO method is 3.63% better on undershoot and 2.99% better on overshoot than PSS-Conv. On the rotor angle with PSS-WOA, a better overshoot value of 46.15% was obtained compared to PSS-CSHO. Meanwhile, at the speed with PSS-WOA, the overshoot value was worse than PSS-CSHO by 62.5%.

V. CONCLUSION AND FUTURE WORKS

In this article, the new improved SHO algorithm is a hybrid between the SHO method and the chaos map method. With this step, the SHO capability is improved. This method is called the CSHO method. CSHO is evaluated and validated with test functions and problems in the real world, namely in the power system. Testing begins by evaluating with the benchmark function. The results obtained were compared with WOA, GOA and SHO. The performance of CSHO shows a better improvement and shows a new balance point between exploration and exploitation. The next evaluation of the proposed CSHO is with real-world problems. In this article, CSHO is applied to obtain PSS parameters that can dampen oscillations optimally. The results obtained were compared with the conventional PSS method (PSS-Conv), PSS using WOA (PSS-WOA), PSS using GOA (PSS-GOA) and PSS using SHO (PSS-SHO). PSS using the CSHO algorithm has the ability to increase the transient stability of the SMIB system and optimal damping characteristics. From the two evaluation methods, it can be concluded that the implementation of the CSHO method shows an algorithm that has a strong approach to optimization problems.

This article uses integration with the chaotic method. This needs to be deepened into the concept of integration using methods that can sharpen the results of the convergence curve and conduct trials with more complex systems.

REFERENCES


