Electrical Power and Energy Systems 77 (2016) 345–352

Contents lists available at ScienceDirect

Electrical Power and Energy Systems

journal homepage: www.elsevier.com/locate/ijepes

Critical Clearing Time prediction within various loads for transient stability assessment by means of the Extreme Learning Machine method

Irrine Budi Sulistiawati [a,b,*], Ardyono Priyadi [a], Ony Asrarul Qudsi [a], Adi Soeprijanto [a], Naoto Yorino [c]

[a] Departement of Electrical Engineering, Institut Teknologi Sepuluh Nopember ITS, Keputih, Sukolilo, Surabaya 60111, Indonesia
[b] Departemen of Electrical Engineering, Institut Teknologi Nasional Malang, Raya Karanglo km 2, Indonesia
[c] Hiroshima University, 1-4-1 Kagamiyama, Higashihiroshima 739-8527, Japan

ARTICLE INFO

Article history:
Received 15 November 2014
Received in revised form 1 October 2015
Accepted 17 November 2015
Available online 11 December 2015

Keywords:
Critical Clearing Time (CCT)
Transient Stability Assessment (TSA)
ELM (Extreme Learning Machine)
Critical trajectory
Load changing

ABSTRACT

The Critical Clearing Time (CCT) is a key issue for Transient Stability Assessment (TSA) in electrical power system operation, security, and maintenance. However, there are some difficulties in obtaining the CCT, which include the accuracy, fast computation, and robustness for TSA online. Therefore, obtaining the CCT is still an interesting topic for investigation. This paper proposes a new technique for obtaining CCT based on numerical calculations and artificial intelligence techniques. First, the CCT is calculated by the critical trajectory method based on critical generation. Second, the CCT is learned by Extreme Learning Machine (ELM). This proposed method has the ability to obtain the CCT with load changes, different fault occurrences, accuracy, and fast computation, and considering the controller. This proposed method is tested by the IEEE 3-machine 9-bus system and Java-Bali 500 kV 54-machine 25-bus system. The proposed method can provide accurate CCTs with an average error of 0.33% for the Neural Network (NN) method and an average error of 0.06% for the ELM method. The simulation result also shows that this method is a robust algorithm that can address several load changes and different locations of faults occurring. There are 29 load changes used to obtain the CCT, with 20 load changes included for the training process and 9 load changes not included.

Introduction

Large disturbances in the rotor angle or Transient Stability Assessment (TSA) plays an important role for electrical power system operations, security, and maintenance. Many researchers have developed methods for obtaining the Critical Clearing Time (CCT) for the transient stability assessment problem, but most of them have proposed direct methods, such as Single Machine Equivalent (SIME), energy function Boundary Controlling Unstable Equilibrium Point (BCU), critical trajectory and artificial intelligence. However, a Time Domain Simulation (TDS) or conventional numerical simulation method are still used to validate the results. The method stated in references [1–8] can accurately provide results because the numerical integration of non linear differential equations is used. However, this approach requires time and needs the detailed process of performing a calculation to guarantee the accuracy. Therefore, it is not suitable for highly dynamic changes, especially for transient stability analysis with variations in the load changes and online assessments.

There is a numerical method, among others, that can be used to calculate the potential energy and kinetic energy for transient stability analysis; this approach is called the energy function method, as stated in reference [5]. This method can quickly provide a transient stability assessment, but it does not guarantee the accuracy of the results. This circumstance means that the energy function method gives only approximate results.

Another method that is believed to be fast in the calculation process and provides accurate results in terms of an exact solution is the critical trajectory method. This method calculates the CCT together with a critical trajectory, and it defines a trajectory that starts from the point of on-fault and ends at a critical condition, such as losing synchronization. The critical trajectory method is a reliable method for analyzing the system stability, especially its transient stability. This method requires a short time in the calculation process and provides accurate results. A trajectory is a critical path that appears when a disturbance appears, and the system is in a critical condition shortly before losing synchronization [9]. This method is an exact method, which uses numerical integration calculations to solve differential equations;

* Corresponding author at: Departement of Electrical Engineering, Institut Teknologi Sepuluh Nopember ITS, Keputih, Sukolilo, Surabaya 60111, Indonesia.

E-mail addresses: irrine10@mhs.ee.its.ac.id (I.B. Sulistiawati), priyadi@ee.its.ac.id (A. Priyadi), adisup@elect-eng.its.ac.id (A. Soeprijanto).

http://dx.doi.org/10.1016/j.ijepes.2015.11.034
0142-0615/

nevertheless, it is sufficiently fast to obtain the CCT. Some new features that modify the end point conditions and the use of a critical generator have been investigated in references [15,17]. The preliminary investigation, which considers the controller, i.e., the Automatic Voltage Regulator (AVR) and governor, has been published in references [18,19].

A transient stability assessment system for determining the Critical Clearing Time (CCT) is developed with the use of artificial intelligence [19–24]. Artificial Intelligence (AI) is used to predict the CCT for the on-line power system. An Artificial Neural Network (ANN) is an advanced calculation process that uses a specific pattern of neurons and weights to solve a problem. A learning or training technique is used in this method [20–22].

Artificial neural networks have the ability to learn the processing of information, such as how the human brain works to determine the critical clearing time in transient stability assessment by changing the weights in neurons. The calculated process indicates that neural networks perform the process of learning or training on previous data, learning complex non linear mappings of the input samples. The result is provided by the mapping weights applied to the input data. In addition, not only can the artificial neural network predict the result for learned data, but it can also provide a satisfactory result for the unlearned data.

The Extreme Learning Machine introduced by Huang et al. [23] is a promising new method of learning compared with Single-Hidden-Layer Feed-forward Network, which utilizes a classical learning approach. ELM not only can make the learning process faster than classical learning but also can provide a small value for the training error. Thus, the performance of this method is superior to other classical methods [23].

This research paper proposes the ELM to obtain the CCT for TSA on the first swing instability. However, the authors believe that this proposed method can also provide an accurate CCT for the multi-swing instability case. Therefore, further investigation is necessary to check the superiority and ability of the proposed method in the near future. In addition, this method is capable of predicting an accurate CCT and requires less calculation time than the other NN [23]. It can also be used for TSA online. The Fouad and Anderson or IEEE 3-machine 9-bus systems and Java-Bali 54-machine 25-bus systems are implemented to validate the proposed method. The various loads and point of faults are also observed to check the superior capability of this proposed method for obtaining CCT. In addition, the CCT is obtained by the critical trajectory method, as stated in references [9–19], for preliminary data.

Basic theory

The fundamental theory plays an important role for this proposed method to obtain the CCT. The proposed method refers to a previous method that is used for the preliminary calculation of obtaining the CCT, and it will also be explained in this section. This section also describes some assumptions that are used in this paper, to make them more easily understood. The previous theory and assumptions will be explained next [17].

Critical Clearing Time

CCT is defined as the maximum time that is allowed to remove the disturbance without interrupting the system's performance. The system will be stable if the disturbance can be cleared before the time allowed. On the other hand, if the system becomes unstable, then the maximum time allowable disturbance cannot be overcome.

A power system must have a Critical Clearing Time that is longer than the operational circuit breaker in the system. Although the CCT is not the main criterion, it should be worked on first when a disturbance occurs. The Critical Clearing Time value is calculated based on the greatest disturbance or the worst case possibility that there is a three-phase short circuit.

There are various methods used to calculate the CCT, such as the energy function, extended equal area criterion, Single Machine Equivalent (SIME), conventional numerical simulation, time domain simulation, and critical trajectory based on losing the synchronism and critical generator. Further development in obtaining the CCT has been performed by the artificial intelligence approach.

Critical trajectory

Based on references [9–19], it is explained that some trajectories affect the behavior of the system before, during, and after a fault occurs. The stability limit of a power system can be explained by utilizing this trajectory. Fig. 1 shows the trajectory of a power system for a single machine system that is connected to an infinite bus with damping.

Here, "1" indicates a fault on the trajectory, "2" indicates a stable condition after a system disturbance, and "4" indicates that the condition is not stable when the disturbance is late-disconnected; "3" is a critical trajectory that is a critical condition of electric power systems.

The theoretical background for obtaining the CCT using the critical trajectory method is stated in references [9–19] and can be explained as follows.

A transient stability condition begins when a disturbance occurs at $t = 0$ after the x_{pre} stable condition. This condition dynamically changes during an interruption $[0, \tau]$ according to the equation:

Here, a "1" indicates a fault on trajectory, "2" indicates a stable condition after a system disturbance, and "4" indicates that the condition is not stable when the disturbance is late-disconnected; "3" is a critical trajectory, which is a critical condition of electric power systems.

The theoretical background for obtaining the CCT using the critical trajectory method is stated in reference [9–19] and can be explained as follows.

A transient stability condition begins when a disturbance occurs at $t = 0$ after the x_{pre} stable condition. This condition is dynamically changing during an interruption $[0, \tau]$ according to the equation:

$$\dot{x} = f_F(x), \quad 0 \leqslant t \leqslant \tau, \quad x(0) = x_{pre} \tag{1}$$

$$x \in R^N, \quad t \in R, \quad f_F : R^N \to R^N \tag{2}$$

The curve "1" is formulated by the equation

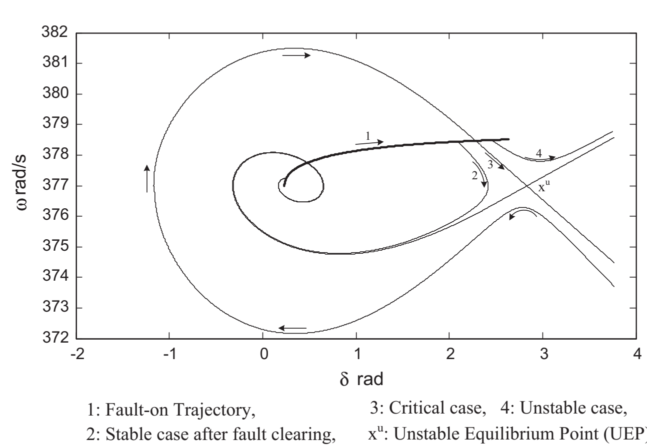

Fig. 1. Trajectory of single machines connected to an infinite bus, with damping.

$$x(t) = X_F(t; x_{pre}), \quad 0 \leqslant t \leqslant \tau \tag{3}$$

$$X_F(\cdot; x_{pre}) : R \to R^N \tag{4}$$

When the disturbance was disconnected at time τ, the conditions will change based on the following equation:

$$\dot{x} = f(x), \quad \tau \leqslant t \leqslant \infty;\ f : R^N \to R^N \tag{5}$$

The curve "2" and "4" are calculated by the equation

$$x(t) = X(t; x^0), \quad \tau \leqslant t \leqslant \infty;\ X(\cdot; x^0 : R^N \to R^N) \tag{6}$$

The curve "3" occurs when the disturbance was disconnected at time τ = CCT, with a note that the initial point x^0 for the critical trajectory is CCT on a fault-on trajectory and is given by the following equation:

$$x^0 = X_F(\tau; x_{pre}), \quad \tau = \text{CCT} \tag{7}$$

Neural network

A neural network can be described as the process of the human brain's neural networks during the training and learning processes. A neural network is potentially applicable as a benchmark for computing nonlinear problems because this method can be used in the absence of a mathematical equation. Therefore, this method is suitable for solving nonlinear problems, especially transient stability analysis in power systems.

The neural network architecture consists of input units (x), weights (w), a hidden layer, and output units. These weights are a key issue for improving the output to attain a target. The output function can be expressed as follows:

$$F(x, w) = f(w_1x_1 + \cdots + w_nx_n) \tag{8}$$

Fig. 2 shows that the standard back-propagation neural networks consist of the inputs, weight, two hidden layers and output. The learning process for this method calculates the weights to obtain the output target. If errors exist, then the weights are updated to improve the solution to make the error of the target small enough. This method is used for comparison of the proposed method.

Proposed method for obtaining CCT

This research paper proposes the Extreme Learning Machine (ELM) method to obtain the predicted Critical Clearing Time. This method is one type of neural network method that has the capability of obtaining the Critical Clearing Time. Therefore, it can provide a timely solution. The performance of this method has also been compared with the single learning algorithm Hidden Layer Feed forward Network (SLFN), which is described in reference [20].

The procedure of the proposed method will be explained as follows:

Fig. 3 shows architecture of the proposed method, which was used to obtain the predicted Critical Clearing Time. This proposed method is derived from SLFN and is called the Extreme Learning Machine (ELM) [22].

For N samples,

$$(x_i, t_i) \tag{9}$$

where $x_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^T \in R^n$ and $t_i = [t_{i1}, t_{i2}, \ldots, t_{im}]^T \in R^m$

The standard SLFN with N hidden nodes and activating function $g(x)$ can be formulated as follows:

$$\sum_{i=1}^{\tilde{N}} \beta_i g_i(x_j) = \sum_{i=1}^{\tilde{N}} \beta_i g(w_i \cdot x_j + b_i) = o_j$$

$$j = 1, \ldots, N, \tag{10}$$

where

$w_i = [w_{i1}, w_{i2}, \ldots, w_{in}]^T$ is the weight vector that connects the ith input to the ith hidden node.

$\beta_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{im}]^T$ is the weight vector that connects the ith hidden node to the output.

b_i is the threshold of the ith hidden node.

w_ix_j = the inner product between w_i and x_j.

Based on the standard SLFN with $\tilde{N}$ hidden nodes and an activation function, $g(x)$ can predict the N samples with zero errors. This outcome means that $\sum_{j=1}^{\tilde{N}} \|o_j - t_j\| = 0$, and thus,

$$\sum_{i=1}^{\tilde{N}} \beta_i g(w_i \cdot x_j + b_i) = t_j, \quad j = 1, \ldots, N \tag{11}$$

This equation can be written in a simpler way, as follows:

$$H\beta = T \tag{12}$$

with

$$H(w_1, \ldots, w_{\tilde{N}}, b_1, \ldots, b_{\tilde{N}}, x_1, \ldots, x_N) \tag{13}$$

$$= \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \cdots & g(w_{\tilde{N}} \cdot x_1 + b_{\tilde{N}}) \\ \vdots & \cdots & \vdots \\ g(w_1 \cdot x_N + b_1) & \cdots & g(w_{\tilde{N}} \cdot x_N + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}} \tag{14}$$

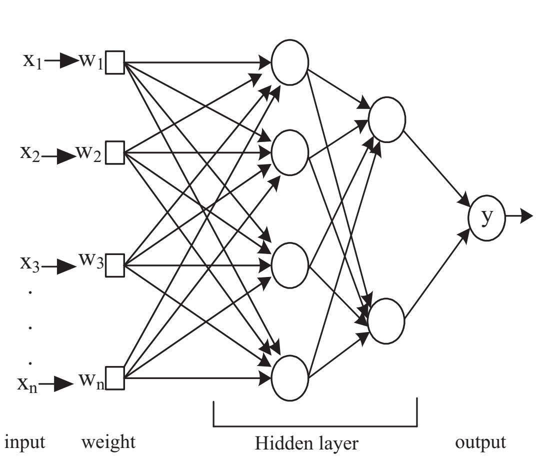

Fig. 2. Neural network architecture with two hidden layers.

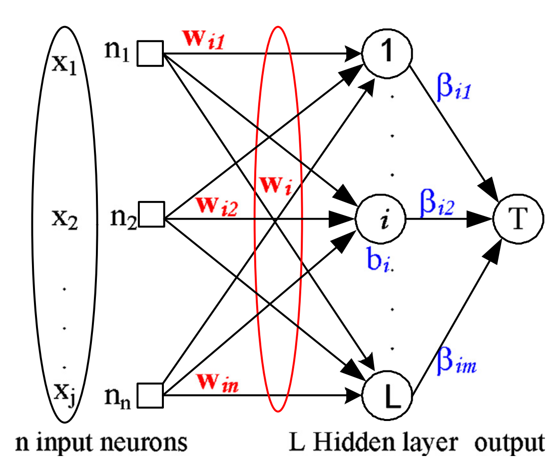

Fig. 3. Architecture of the Extreme Learning Machine.

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix}_{\tilde{N}\times m} \quad \text{and} \quad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{n\times m}$$

H is the hidden layer output matrix of the neural network, and $x_1, x_2, \ldots, x_N$ is the ith hidden node output with respect to the inputs. T is the target or output. The proposed method does not require a bias for the hidden layer (b_i) and tuning for the input weights (w_i). The weights on the hidden layer output matrix are obtained at random without any training iterations. The output weights are determined by the formula $H\beta = T$, and this relationship is linearized by least-squares for the linear systems, with

$$\hat{\beta} = H^{\dagger}T \tag{15}$$

Problem formulation

Power system model

Multi-machine system

A multi-machine system is defined as a model xd' with generators that are indicated by two differential equations. This model is called a classical swing equation and can be represented as follows:

$$\begin{aligned} M_i\dot{\omega}_i &= P_{mi} - P_{ei}(\delta) - D_i\omega_i \\ \dot{\delta}_i &= \omega_i \end{aligned} \tag{16}$$

The multi-machine systems used centre of angle (COA) swing equations that can be written as follows:

$$M_i\dot{\tilde{\omega}}_i = P_{mi} - P_{ei}(\theta) - \frac{M_i}{M_T}P_{COA} - D_i(\tilde{\omega}_i) \tag{17}$$

$$\dot{\theta}_i = \tilde{\omega}_i \tag{18}$$

where

$$M_T = \sum_{i=1}^{n} M_i; \quad \omega_0 = \frac{1}{M_T}\sum_{i=1}^{n} M_i\omega_i; \quad \delta_0 = \frac{1}{M_T}\sum_{i=1}^{n} M_i\delta_i;$$
$$\theta_i = \delta_i - \delta_0; \quad \tilde{\omega}_i = \omega_i - \omega_0;$$
$$P_{COA} = \sum_{i=1}^{n}(P_{mi} - P_{ei}(\theta)); \;\; P_{e_i}(\theta) = \sum_{j=i}^{n} Y_{ij}E_iE_j\sin(\theta_i - \theta_j + \alpha_{ij})$$

P_{mi} is the ith mechanical power input;
ω_i is the ith generator rotor speed;
δ_i is the ith generator angle deviation;
M_i is the ith moment of inertia;
D_i is the ith damping coefficient;
E_i is ith voltage behind the transient reactance;
P_{ei} is the ith electric power.

AVR and Governor are represented as follows:

$$\dot{E}_i = \frac{1}{T_{AVRi}}\left[(E_{0i} - E_i) + K_{AVRi}(V_{refi} - V_{ti})\right] \tag{19}$$

$$\dot{P}_{mi} = \frac{1}{T_{GOV_1}}\left[(P_{mrefi} - P_{mi}) + K_{GOV}(\tilde{\omega}_i)\right] \tag{20}$$

with

$$P_{ei}(\delta) = \sum_{j=i}^{n} Y_{ij}E_iE_j\cos(-\delta_i + \delta_j + \alpha_{ij}) \tag{21}$$

The CCT obtained by the Critical Trajectory method has been published before and can be explained with the following formulation:

$$\min_{x^0, x^1, \ldots, x^{m+1}, \varepsilon, \tau}\left\{\sum_{k=0}^{m}(\mu^k)'(\mu^k) + (\mu^{m+1})'W(\mu^{m+1})\right\} \tag{22}$$

where

$x_k \in R^N, \; (k = 0, \ldots, m), \quad \varepsilon \in R, \quad \tau \in R$

$$\mu^k = x^{k+1} - x^k - \frac{\dot{x}^{k+1} + \dot{x}^k}{|\dot{x}^{k+1} + \dot{x}^k|}\varepsilon \tag{23}$$

$$\dot{x}^k = f(x^k) \tag{24}$$

with the following boundary conditions:

$$x^0 = X_F(\tau, x_{pre}) \tag{25}$$

$$\mu^{m+1} = x^{m+1} - x^u \quad with \quad f(x^u) = 0 \tag{26}$$

This minimizing problem is solved by The Newton–Raphson method to obtain the critical trajectory together with the CCT. Then, this procedure is repeated to obtain the CCT for all of the points of the faults, when varying the load. This CCT is learned by the proposed method to speed up the calculation that will be applied online. However, not all of the obtained CCTs for varying the loads are learned by the proposed method; only some of them are, specifically, 31%. The aim is to provide robustness in the proposed method.

The proposed method

The proposed method is called the Extreme Learning Machine (ELM), and it has the ability to select random hidden nodes and determine the output weights analytically [22–24]. The proposed method is also capable of providing results in less time during the learning process. In this paper, the author rigorously proves that the input weights and hidden layer biases of the SLFNs can be randomly assigned. If the activation functions for the hidden layer are infinitely differentiable after the input weights and hidden layer biases are chosen randomly, then the SLFNs can be simply considered to be a linear system. The output weights (which link the hidden layer to the output layer) of the SLFNs can also be analytically determined through a simple generalized inverse operation. This Process can still give good results in solving problems that involve a large and complex system. In addition, the Extreme Learning Machine (ELM) has a learning speed that can be thousands of times faster than the traditional feed forward network learning algorithms, such as the back-propagation neural network (BPNN) algorithm, while the obtained result is better in terms of the general performance.

The algorithm of the training process of the ELM is based on the Single Layer Feed forward Network (SLFN) and has an efficient and simple algorithm that can be written as follows:

Find specific $\hat{w}_i, \; \hat{b}_i, \; \hat{\beta} \; (i = 1, \ldots, \tilde{N})$

$$\begin{aligned} &\left\|H(\hat{w}_1, \ldots, \hat{w}_{\tilde{N}}, \hat{b}_1 \ldots, \hat{b}_{\tilde{N}})\hat{\beta} - \mathrm{T}\right\| \\ &\quad = \min_{w_i b_i \beta}\left\|H(w_{1,\ldots,w_{\tilde{N}},b_1,\ldots,b_{\tilde{N}}})\beta - \mathrm{T}\right\| \end{aligned} \tag{22}$$

Minimizing the cost function given by

$$E = \sum_{j=1}^{N}\left(\sum_{i=1}^{\tilde{N}}\beta_i g(w_i \cdot x_j + b_i) - t_j\right)^2 \tag{23}$$

Gradient-based learning algorithms denoted as H are used to search for the minimum of $\|H\beta - \mathrm{T}\|$. The procedure tends to be simpler because the vector W becomes the set of weights (w_i, β_i) and bias (b_i) parameters, and it is iteratively adjusted as

$$W_k = W_{k-1} - \eta\frac{\partial E(W)}{\partial W} \tag{24}$$

Here, η is the learning rate.

The input weight w_i and the hidden layer biases b_i can be fixed because, in fact, they both are not necessarily tuned, and the hidden layer output matrix H actually remains unchanged when the learning process begins [24]. Thus, we find solution $\hat{\beta}$ of the linear system $H\beta = T$

$$\left\| H(\hat{w}_1, \ldots, \hat{w}_{\tilde{N}}, \hat{b}_1 \ldots, \hat{b}_{\tilde{N}})\hat{\beta} - T \right\| = \min_{\beta} \left\| H(w_{1,\ldots,w_{\tilde{N}},b_1,\ldots,b_{\tilde{N}}})\beta - T \right\| \tag{25}$$

Simulation results

Simulation procedure

The proposed method is tested to obtain the CCT using the IEEE3-machine 9-bus system and the Java-Bali 500 kV 54-machine 25-bus system, as shown in Figs. 4 and 5, respectively.

The first simulation is on the IEEE 3-machine 9-bus system, as shown in Fig. 4, and it can be explained as follows: bus 6 is assumed to have changing loads. Generator 1 is a hydro electric plant, while 2 and 3 are the steam generators. Simulation is accomplished by performing a three-phase short circuit at points A, B, and G. In addition, the transmission line is assumed to have double circuits and a point of fault that occurs close to the bus.

A three-phase short circuit fault is given at one point, and then, CCT is calculated on this condition. The obtained CCT is repeated at each changing load at bus 6. In addition, the calculation of the CCT is repeated by three-phase short circuit faults at different points. The fault points A, B, and G are investigated. These three points of the fault can be explained below:

(a) Fault "1"or fault point A is the point of fault between bus 1 and 4 and is close to bus 1.
(b) Fault "2"or fault point B is the point of fault between bus 2 and 7 and is close to bus 2.
(c) Fault "3"or fault point G is the point of fault between bus 7 and 8 and is close to bus7.

The second simulation is the Java-Bali 54-machine 25-bus system, as shown in Fig. 5. Bus 15 is a bus that gradually changes load. At each change in the load, a three-phase short circuit simulation is performed to obtain the same Critical Clearing Time as before. It is assumed that a three-phase short circuit occurs at the fault points B, C, and G. Fault point B is the bus between Cibinong and Bekasi, fault point C is the bus between Saguling and Cirata, and fault point G is the bus between Cirata and Cibatu. Other assumptions are the same as the previous simulation of the power system model.

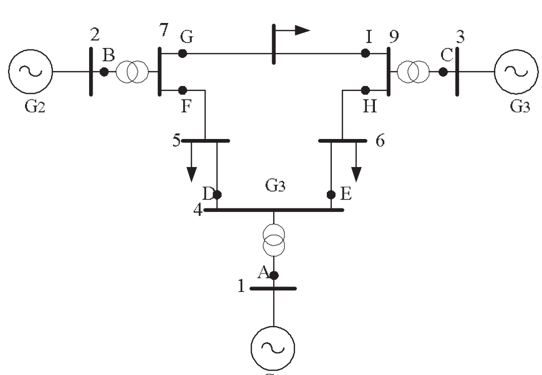

Fig. 4. IEEE 3-machine 9-bus system.

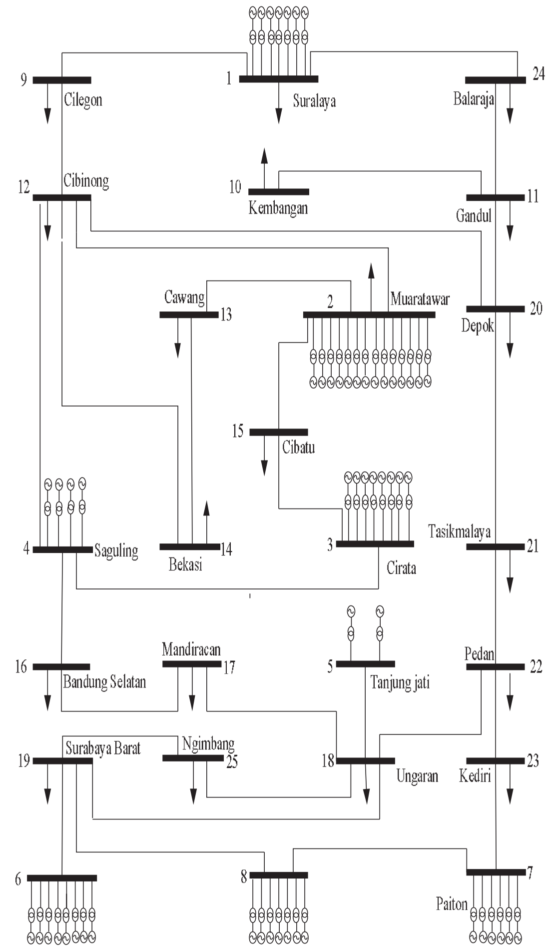

Fig. 5. Java Bali 500 kV 54-machine 25-bus system.

Obtaining CCT

A numerical simulation is performed by the critical trajectory method, which is performed using Eqs. (6), (7) and stated in the references [9]. The simple AVR and governor models are used for both power system models, as stated in references [18,19]. A further feature is used to obtain the CCT with load variations. This approach is to determine the daily load profile in the electric power system. The conventional numerical simulation method is used to validate obtaining the CCT.

The next step is the learning process while using the proposed method. The obtaining of the CCT is calculated by the proposed method and is performed approximately 200 times, to validate the robustness and accuracy.

Figs. 6 and 7 illustrate the obtaining of the CCT using the two methods: the NN method and the proposed method called ELM. These iterations are run 200 times. It is shown that the proposed method can provide a similar CCT in a number of iterations, which is not the case in other methods. This finding proves that the proposed method can provide a robust result although it is a statistical method.

Tables 1-6 show the simulation results that were performed by the critical trajectory (CT), back-propagation Neural Network (NN), and Extreme Learning Machine (ELM) for the IEEE 3-machine 9-bus system and Java Bali 500 kV 54-machine 25-bus system. The load variations are also listed in the tables. Tables 1 and 2 show obtaining the CCT in seconds using CT, NN, and ELM. Fault variations are also listed. The obtained CCT numbers using the proposed method are similar to using the critical trajectory method. It is observed

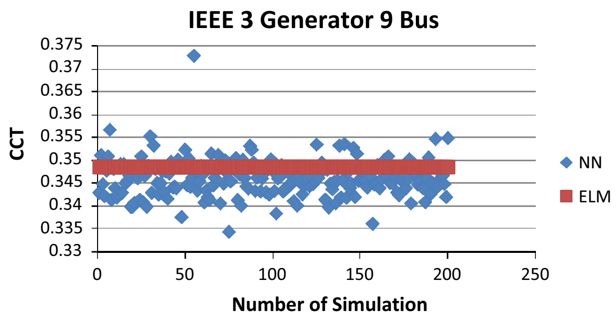

Fig. 6. CCT for IEEE 3-machine 9-bus system simulated 200 times.

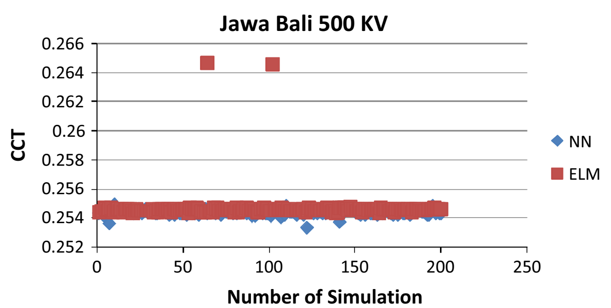

Fig. 7. CCT for Java Bali 500kV54-machine 25-bus system simulated 200 times.

Table 1a
Obtaining CCT for IEEE 3-machine 9-bus system for Fault-1.

Load variations		Obtaining CCT for Fault-1		
P (MW)	*Q* (MVar)	CT (s)	NN (s)	ELM (s)
95	35	0.3485	0.3512	0.3487
105	45	0.3635	0.3631	0.3633
115	55	0.3805	0.3884	0.3802
125	65	0.3995	0.4050	0.3979
135	75	0.4205	0.4213	0.4197
145	85	0.4445	0.4425	0.4441
155	95	0.4715	0.4815	0.4735

Table 1b
Obtaining CCT for IEEE 3-machine 9-bus system for Fault-2.

Load variations		Obtaining CCT for Fault-2		
P (MW)	*Q* (MVar)	CT (s)	NN (s)	ELM (s)
95	35	0.2145	0.2148	0.2151
105	45	0.2165	0.2166	0.2163
115	55	0.2195	0.2199	0.2194
125	65	0.2215	0.2217	0.2215
135	75	0.2245	0.2234	0.2245
145	85	0.2265	0.2270	0.2265
155	95	0.2295	0.2293	0.2289

Table 1c
Obtaining CCT for IEEE 3-machine 9-bus system for Fault-3.

Load variations		Obtaining CCT for Fault-3		
P (MW)	*Q* (MVar)	CT (s)	NN (s)	ELM (s)
95	35	0.2335	0.2347	0.2341
105	45	0.2375	0.2373	0.2368
115	55	0.2405	0.2418	0.2406
125	65	0.2435	0.2447	0.2439
135	75	0.2475	0.2483	0.2470
145	85	0.2505	0.2495	0.2509
155	95	0.2535	0.2535	0.2533

Table 2a
Obtaining CCT Fault-1 for the Java Bali 500 kV 54-machine 25-bus system.

Load variations		Obtaining CCT for Fault-1		
P (MW)	*Q* (MVar)	CT (s)	NN (s)	ELM (s)
1162	355	0.6653	0.6678	0.6542
1187	340	0.6904	0.6936	0.6862
1207	360	0.7135	0.7159	0.7170
1232	385	0.7473	0.7402	0.7537
1252	405	0.7785	0.7839	0.7769
1272	425	0.8128	0.8076	0.8043
1277	430	0.8219	0.8128	0.8128
1297	450	0.861	0.8388	0.8528
1302	455	0.8716	0.8319	0.8634

Table 2b
Obtaining CCT Fault-2 for the Java Bali 500 kV 54-machine 25-bus system.

Load variations		Obtaining CCT for Fault-2		
P (MW)	*Q* (MVar)	CT (s)	NN (s)	ELM (s)
1162	355	0.2546	0.2545	0.2546
1187	340	0.2557	0.2554	0.2556
1207	360	0.2563	0.2564	0.2561
1232	385	0.2571	0.2570	0.2570
1252	405	0.2579	0.2580	0.2572
1272	425	0.2587	0.2587	0.2577
1277	430	0.2589	0.2588	0.2579
1297	450	0.2596	0.2592	0.2596
1302	455	0.2598	0.2591	0.2601

Table 2c
Obtaining CCT Fault-3 for the Java Bali 500 kV 54-machine 25-bus system.

Load variations		Obtaining CCT for Fault-3		
P (MW)	*Q* (MVar)	CT (s)	NN (s)	ELM (s)
1162	355	0.1918	0.1906	0.1923
1187	340	0.1918	0.1914	0.1924
1207	360	0.1921	0.1917	0.1922
1232	385	0.1925	0.1926	0.1933
1252	405	0.1923	0.1921	0.1938
1272	425	0.193	0.1934	0.1938
1277	430	0.193	0.1937	0.1938
1297	450	0.1929	0.1928	0.1946
1302	455	0.1931	0.1931	0.1951

Table 3a
Error of CCT Fault-1 for IEEE 3-machine 9-bus system.

Load variations		Obtaining CCT for Fault-1	
P (MW)	*Q* (MVar)	Neural Network (s)	Extreme Learning Machine (s)
95	35	0.0027	0.0002
105	45	0.0004	0.0002
115	55	0.0079	0.0003
125	65	0.0055	0.0016
135	75	0.0008	0.0008
145	85	0.002	0.0004
155	95	0.01	0.002

that the proposed method can provide an acceptable CCT compared to other methods. This finding means that the proposed method has the potential to become an alternative method for obtaining the CCT.

Tables 3 and 4 show the error in CCT in seconds, for the proposed method and NN compared with the critical trajectory method. The maximum error is 0.0238 in seconds. It is observed that the proposed method can obtain an accurate CCT for all of the load variations and fault points in both systems.

Table 3b
Error of CCT Fault-2 for the IEEE 3-machine 9-bus system.

Load variations		Obtaining CCT for Fault-2	
P (MW)	*Q* (MVar)	Neural Network (s)	Extreme Learning Machine (s)
95	35	0.0003	0.0196
105	45	0.00001	0.0203
115	55	0.0004	0.0211
125	65	0.0002	0.0224
135	75	0.0011	0.0225
145	85	0.0005	0.0244
155	95	0.0002	0.0238

Table 3c
Error of CCT Fault-3 for the IEEE 3-machine 9-bus system.

Load variations		Obtaining CCT for Fault-3	
P (MW)	*Q* (MVar)	Neural Network (s)	Extreme Learning Machine (s)
95	35	0.0012	0.0006
105	45	0.0002	0.0007
115	55	0.0013	0.0001
125	65	0.0012	0.0004
135	75	0.0008	0.0005
145	85	0.001	0.0004
155	95	0.002	0.0002

Table 4a
Error of CCT Fault-1 for the Java Bali 500 kV 54-machine 25-bus system.

Load variations		Obtaining CCT for Fault-1	
P (MW)	*Q* (MVar)	Neural Network (s)	Extreme Learning Machine (s)
1162	355	0.0025	0.0111
1187	340	0.0032	0.0042
1207	360	0.0024	0.0035
1232	385	0.0071	0.0064
1252	405	0.0054	0.0016
1272	425	0.0052	0.0085
1277	430	0.0091	0.0091
1297	450	0.0222	0.0082
1302	455	0.0397	0.0082

Table 4b
Error of CCT Fault-2 for the Java Bali 500 kV 54-machine 25-bus system.

Load variations		Obtaining CCT for Fault-2	
P (MW)	*Q* (MVar)	Neural Network (s)	Extreme Learning Machine (s)
1162	355	0.0001	0
1187	340	0.0003	0.0001
1207	360	0.0001	0.0002
1232	385	0.0001	0.0001
1252	405	0.0001	0.0007
1272	425	0	0.001
1277	430	0.0001	0.001
1297	450	0.0004	0
1302	455	0.0007	0.0003

Table 4c
Error of CCT Fault-3 for the Java Bali 500 kV 54-machine 25-bus system.

Load Variations		Obtaining CCT for Fault-3	
P (MW)	*Q* (MVar)	Neural Network (s)	Extreme Learning Machine (s)
1162	355	0.0012	0.0005
1187	340	0.0004	0.0006
1207	360	0.0004	0.0001
1232	385	0.0001	0.0008
1252	405	0.0002	0.0015
1272	425	0.0004	0.0008
1277	430	0.0007	0.0008
1297	450	0.0001	0.0017
1302	455	0	0.002

Table 5
CPU calculation time for the ieee 3-machine 9-bus system.

Load variations		Calculation time		
P (MW)	*Q* (MVar)	CT (s)	NN (s)	ELM (s)
95	35	0.7584	0.3588	0.0022
105	45	0.8154	0.3008	0.0022
115	55	0.8133	0.3120	0.0044
125	65	0.8034	0.3276	0.0044
135	75	0.7931	0.3187	0.0044
145	85	0.7975	0.3299	0.0067
155	95	0.7995	0.3120	0.0044
Average		0.7972	0.3228	0.0033

Table 6
CPU calculation time for the Java Bali 500 kV 54-machine 25-bus system.

Load variations		Calculation time		
P (MW)	*Q* (MVar)	CT (s)	NN (s)	ELM (s)
1162	355	0.9385	0.6708	0.1699
1187	340	0.9422	0.3114	0.1681
1207	360	0.9214	0.2045	0.1664
1232	385	0.9354	0.2392	0.1595
1252	405	0.9337	0.2333	0.1768
1272	425	0.9374	0.2111	0.1681
1277	430	0.9304	0.2333	0.1837
1297	350	0.9483	0.2778	0.1629
1302	455	0.9482	0.2333	0.1681
Average		0.9373	0.2905	0.1692

Tables 5 and 6 show the calculation time for the CT, NN, and ELM methods in seconds. The average of the calculation time is also shown. The proposed method is 1.72 times faster than NN and 5.54 times faster than CT. This finding means that the proposed method is fast enough to obtain the CCT and be potentially applicable for online transient stability assessment.

Conclusions

The proposed method is one type of intelligent technique that can obtain an accurate and robust CCT. The maximum error is 0.0238 in seconds for both systems tested.

The proposed method can also quickly calculate the CCTs, which are 5.54 and 1.72 times faster compared to the CT and NN method, respectively. Therefore, the proposed method is potentially applicable for online transient stability assessment. As an additional feature, it can obtain the CCT while considering the controller (governor) and AVR.

Acknowledgements

The authors gratefully acknowledge the contributions of the Directorate General of Higher Education in Indonesia (DIKTI) for the support of this work

References

[1] Yorino N, Sasaki H, Tamura Y, Yokoyoma YR. Generalized analysis method of auto-parametric resonances in power systems. IEEE Trans Power Syst 1989;A4 (3):1057–64.
[2] Vega DR, Erns D, Fereira CM, Pavella M. A contingency filtering ranking and assessment technique for on-line transient stability studies. In: International conference on electric utility deregulation and restructuring and power technologies; 2000.
[3] Fuji W, Wakisaka J, Iwamoto S. Transient stability analysis based on dynamic single machine equivalent. In: 39th North American power symposium; 2007.
[4] Haque MH. Further developments of the equal-area criterion for multi-machine power system. Electr Power Syst Res 1995.
[5] Chiang HD, Wu FF, Varaiya P. A BCU method for direct analysis of power system transient stability. IEEE Trans Power Syst 1994;9:1194–208.

[6] IEEE Power Engineering Society. Inter-area oscillations in power systems. System Dynamic Performance Subcommittee Special Publication 95TP 101; 1995.
[7] IEEE Power Engineering Society. Voltage stability assessment: concepts practices and tools. Power System Stability Subcommittee Special Publication SP101PSS; 2003.
[8] Zarate-Minano R, Van Cutsem T, Milano F, Conejo AJ. Securing transient stability using time-domain simulations within an optimal power flow. IEEE Trans Power Syst 2010;25(1):243–53.
[9] Yorino N, Saito T, Li HQ, Kamei Y, Sasaki H. A new method for transient stability analysis. In: The papers of technical meeting on power system technology, IEE Japan, 2003, no. PE-03-83, PSE-03-94 [in Japanese].
[10] Yorino N, Kamei Y, Zoka Y. A new method for transient stability assessment based on critical trajectory. In: Proceedings of the 15th Power System Computation Conference (PSCC), 2005, Paper ID 20-3.
[11] Yorino N, Priyadi A, Zoka Y, Yasuda H, Kakui H. A novel method for transient stability analysis as boundary value problem. In: Proceedings of the International Conference on Electrical Engineering (ICEE), Okinawa, July, 2008.
[12] Priyadi A, Yorino N, Eghbal M, et al. Transient stability assessment as boundary value problem. In: IEEE proceedings on annual electrical power and energy conference, Vancouver, Canada, October 2008; 1–6.
[13] Yorino N, Priyadi A, Zoka Y, et al. Transient stability analysis as boundary value problem. In: Proceedings of the XI Symposium of Specialists in Electric Operational and Expansion Planning (XI SEPOPE) Belem, PA, March 2009, SP-032. p. 1–11.
[14] Yorino N, Priyadi A, Kakui H, Takeshita M. A new method for obtaining critical clearing time for transient stability. IEEE Trans Power Syst 2010;25(3):1620–6.
[15] Yorino, N., Priyadi, A., Ridzuan, B.A.M., Sasaki, Y., Zoka, Y., Sugihara, H., A novel method for direct computation CCT for TSA using critical generator conditions, TENCON, Fukuoka, Japan, 23 November 2010. p. 1–6.
[16] Priyadi A, Yorino N, Sasaki Y, Tanaka M, Fujiwara T, Zoka Y, et al. Comparison of critical trajectory methods for direct CCT computation for transient stability. IEEJ Trans Power Energy 2010;130(10):870–6.
[17] Priyadi A, Yorino N, Tanaka M, Fujiwara T, Zoka Y, Kakui H, et al. A direct method for obtaining critical clearing time for transient stability using critical generator conditions. Eur Trans Electr Power 2012;22(5):674–87.
[18] Yorino N, Priyadi A, Zoka Y, Sasaki Y, Sugihara H. A new method for direct computation of critical clearing time for transient stability analysis. In: Proc on IREP, Rio De Janeiro, Brazil, August 2010. p. 1–9.
[19] Yorino N, Priyadi A, Ridzuan BAM, Sasaki Y, Zoka Y, et al. Direct computation of critical clearing time for transient stability analysis. In: Proceeding on 17th PSCC, Stockholm, Sweden, August 22–26, 2011.
[20] Lin YJ. Explaining critical clearing time with the rules extracted from a multilayer perceptron artificial neural network. J Electr Power Energy Syst 2010;32.
[21] Karami A. Power system transient stability margin estimation using neural network. J Electr Power Energy Syst 2011;33:983–91.
[22] Huang GB, Zhu QY, Siew CK. Extreme learning machine: a new learning scheme of feed-forward neural networks. In: Proceedings of international joint conference on neural networks; 2004.
[23] Huang GB, Zhu QY, Siew CK. Extreme learning machine: theory and application. J Sci Direct Neuro-comput 2006;70:489–501.
[24] Zhu QY, Qin AK, Suganthan PN, Huang GB. Evolutionary extreme learning machine. J Pattern Recogn Soc 2005;38:1759–63.