ERJ
Engineering Research Journal
Faculty of Engineering
Minoufiya University

REPLACEMENT OF CONVENTIONAL PSS WITH A GENETIC TUNED PID CONTROLLER BASED ON DIFFERENT SEARCH CRITERION

A. M. Abdel-Hamid

Department of Electrical Engineering, Faculty of Engineering
Minoufiya University, Shebin El-Kom, Egypt.

ABSTRACT

This paper proposes a genetically tuned PID controllers for power system models with uncertain parameters to replace the conventional power system stabilizer. This may overcome the problems arising due to the fact that PSS is only effective for a linear power system model with certain parameters and disturbances with relatively limited small range around the nominal operating condition. The first step in the design procedure is to find out appropriate PID parameters which are essential to assess and initiate the genetic search within a confident evolution environment. The industrially recognized Ziegler-Nicholes methodology will be employed for this purpose. Secondly, different search criteria such as Integral of Time multiplied by Absolute Error (ITAE), Integral of Absolute Magnitude of the Error (IAE), Integral of the Square of the Error (ISE), and Mean of the Square of the Error (MSE) are implemented to ensure the robustness of the proposed controller. Several experiments will be undertaken to evaluate which of these four performance criteria produce the best results when used in conjunction with a Genetic Algorithm (GA). The Results of implementing the proposed GA-tuned PID controller show that the most satisfactory response, will be obtained if a GA with MSE or ISE criterion is selected to tune the PID controller.

يقدم هذا البحث مقترحا لاستبدال موازنات نظم القوى الكهربية التقليدية بحاكمات تناسبية تفاضلية تكاملية PID يتم تنغيمها بطريقة وراثية Genetically Tuned وذلك لتناسب نماذج حقيقية لنظم القوي الكهربية والتي تتميز عمليا بعدم تأكيدية معاملاتها Parameter Uncertainties وقد يؤدي هذا المقترح للتغلب علي المشاكل التي تواجهها الموازنات التقليدية والمصممة على فرضية نماذج خطية لنظم القوى الكهربية والتي يتم اختبار فاعليتها فقط على مدى محدود من أخطاء صغيرة Small Disturbances حول ظروف التشغيل المقننة للنظام مع تجاهل عدم تأكيدية المعاملات في تصميم تلك الموازنات. وكخطوة أولي لتصميم الحاكم المتناغم وراثيا Genetically Tuned يتم إيجاد قيم مبدئية لمعاملات هذا الحاكم من أجل بدء وتقييم عملية الاختيار الوراثية لمعاملات تناغمية Genetically Tuned PID Coefficients ضمن بيئة تطوير يمكن الوثوق في مدي الاختيار المميز لها Confident Evolution Environment و لتحقيق هذه الخطوة تم تطبيق طريقة تسيجلر- نيكولز Ziegler-Nicholes والمتعارف علي مميزاتها العملية في التصميم التقليدي للحاكمات. أما الخطوة الثانية فتمثل مقترحا لاختبار أربعة معايير مختلفة للبحث والاختيار الوراثي من اجل الوصول للتنغيم الأحسن لمعاملات الحاكم التناسبي التفاضلي التكاملي والذي بدوره يعمل بتناغم خلال مدي واسع من عدم تأكيدية معاملات نظم القوي الكهربية. المعايير المختلفة المقترح اختبارها هي التكامل الزمني مضروبا في الخطأ المطلق Integral of Time Multiplied by Absolute Error (ITAE) وتكامل الخطأ المطلق Integral of Absolute Magnitude of the Error (IAE) وتكامل مربع الخطأ Integral of the Square of the Error (ISE) ومتوسط مربع الخطأ Mean of the Square of the Error (MSE) هذا وقد تم اختبار تلك المعايير علي نظام قوي كهربي يفترض في معاملاته عدم تأكيدية تتراوح بين (±٢٠~±٥٠)% وقد بينت النتائج تفوق طريقتي متوسط مربع الخطأ وتكامل مربع الخطأ كأحسن معيارين يؤديان إلى تناغم الحاكم بما يضمن متانة في الأداء Robustness مع أحسن مواصفات اتزان Stability Indexes.

Keywords: Power system stabilizer (PSS); Ziegler-Nicholes PID controller design; Genetic Algorithm.

1. INTRODUCTION

Power system stabilizers are widely used [1-8] to improve the system steady state stability, i.e. the performance of synchronous generator under disturbance conditions. PSS is considered as a feedback controller connected to a power generating unit. The purpose of stabilizing signal from PSS is to hinder poor-damped machine speed electro-mechanical oscillations. These oscillations arise basically due to the machine pair of complex poles with positive or very small negative real parts in a linearized model. There are several kinds of PSS, and some, while properly achieve damping of the system swings, may introduce negative damping at outlying frequencies. This may result in serious situations when interacting with generator shaft torsional oscillations and with other systems [8]. The most widely employed stabilizer is in the form of lead-lag compensator in which the gain settings and time constants are fixed at certain values determined under particular operating condition [1,9]. However, it is easy to prove the incompetence of such PSS structure if the power system is exposed to high disturbances or if one takes the uncertainty of power system parameters in concern [9,18].

In fact there is a permanent drifting of the operating conditions of a power system due to the continuous load changes or unpredictable major disturbances. Accordingly, several control schemes has been proposed to adapt the stabilizer parameters in order to maintain good dynamic performance over a wide range of operating Conditions [10,11,12,14,18] .In [11] a self tuning PI is added to the excitation system forming a PSS to enhance the damping of the low frequency electromechanical oscillation characterizing generator rotor dynamic behavior. It is clear from the results of the proposed system that there is insufficient damping, while original PSS without such tuned controller gives better response. This is due to the fact that the proposed tuned system is reduced to just a second-order equation while the traditional PSS is of higher order. In [11] the authors have proposed an optimal design of power system dynamic stabilizer by grey prediction PID control. However, the grey prediction is implemented to find out which control signal suits a specific generator in a multi-machine system; in addition the algorithm also searches the optimal parameters of the PID controller. Although results of this algorithm seems to be fast and sufficient, the used grey algorithm gathers Integral of Absolute Magnitude of the Error (IAE), Overshoot (OS), and Steady State Error (SSE) to form the genetic fitness function from three functions; g1, g2, and g3 corresponding to IAE, OS, and SSE respectively. This may add more complexity to the algorithm, specially, for the procedure of on-line implementation.

Nowadays, with the development in digital technology, it has become possible to develop and implement new controllers based on modern and more sophisticated synthesis techniques. Indeed, controllers based on robust optimal control, adaptive control, artificial intelligence (Fuzzy logic, neural networks, and genetic algorithms) are being developed [10-16]. To achieve an efficient fuzzy logic controller, several steps such as the selection of control variables, the membership function definition, the rule creation, the fuzzy inference and the defuzzification strategy should be carefully considered [9,13]. Genetic Algorithms (GA) have been shown to be capable of locating high performance controllers in complex domains without experiencing the difficulties associated with high dimensionality or false optima as may occur with other optimization techniques [11,12]. Moreover, Genetic Algorithms (GA) as a stochastic global search method can imitates the process of natural evolution. Consequently, this paper presents genetic algorithms to perform online tuning of a PID controller that would be evaluated for the system every time. In order to guarantee the best results as well as to ensure the robustness of the proposed controller, different search criteria are implemented.

2. THE PROPOSED SYSTEM MODEL

2.1. The Power System Model

Fig. 1 shows a schematic and block diagram of the system under study. This figure shows a power system which is composed of a single machine connected to an infinite bus via a transmission line.

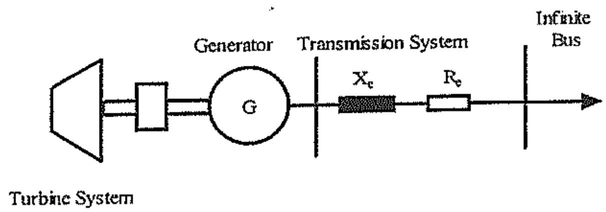

Fig. 1. The Power System Model

Fig. 2 shows the well known de'Mello and Concordia [1,6,17] linearized model of a single machine system. The details of this model can be founded by reference to their paper [1] and Appendix 2.

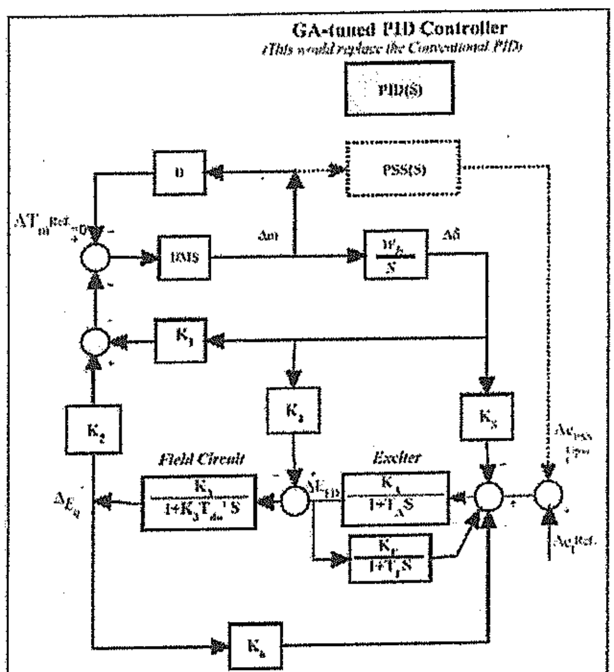

Fig. 2. Linearized Model of the Power System (Phillips –Heffron)

2.2. Conventional PSS

The most widely used PSS is in the form of lead-lag compensator in which the gain settings and time constants are fixed at certain values determined under particular operating condition. Simulation results of such PSS have proven to be very acceptable if the operating conditions and the system data remain unchanged. Unfortunately, it is a well known feature of power systems that the operating conditions are always varying. Also the parameters of the power system are very sensible to any change in the system components. Fig. 3 shows the system response if a light change in these parameters is assumed (Only 5% change in S^3 coefficient in the overall Transfer Function given in Appendix 5). It is clear that the PSS alone can't hinder these changes. In this figure only one parameter is assumed to change.

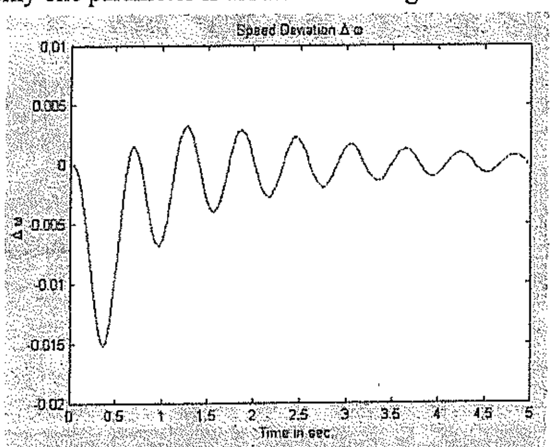

Fig. 3. The system closed loop step response (PSS with light change in system parameters)

3. THE PROPOSED GENETICALLY TUNED PID CONTROLLER

3.1. Genetic Algorithm Tuning Procedure

The aim of this paper is to find well tuned PID parameters that achieve power system stability for any disturbance irrespective of the parameters changes or uncertainty. To solve this problem, one looks for some solution, which will be the best among others. The space of all feasible solutions (it means objects among those the desired solution is, e.g. PID controller parameters in our case) is called search space. Each point in the search space represents one feasible solution. Each feasible solution can be "marked" by its value or fitness for the problem. The goal solution is, actually, one point (or more) among feasible solutions - that is one point in the search space. The looking for a solution is then equal to a looking for some extreme (minimum or maximum) in the search space. The search space can be whole known by the time of solving a problem, but, usually, only a few points from it are known while other points should be generated as the process of finding solution continues. Genetic Algorithm as a stochastic search heuristic, inspired by biological evolution, has proved to be very efficient to find out such goal solution. The sequence of the necessary steps required to find out a solution by GA can be summarized as shown in Fig.4. The basic GA algorithm involves the generation of a population of possible solutions, evaluation of the solutions according to a fitness function, selection of a set of fit "parent" solutions, and finally reproduction of those parents to generate a new population of possible solution.

The important processes which are simulated during a GA include:

(1) Representation and Fitness
(2) Generation of an initial population
(3) Selection of solutions
(4) Reproduction

Note that the terminology of GA remains firmly originated to biology, and so it is common to discuss "parent", "child", "offspring", "chromosomes", and so on. The above processes are explained in brief in Appendix 8.

3.2. Performing the Genetic Algorithm

The genetic algorithm is accomplished via implementing the Matlab GA Toolbox using the following command:

```
%Iterating the genetic algorithm
[x,endPop,bPop,traceInfo]=ga(bounds,evalFN,evalOps,startPop,opts,...
termFN,termOps,selectFN,selectOps,xOverFNs,xOverOps,mutFNs,mutOps);
```

Once the above Matlab command "ga" is entered, the genetic algorithm will iterate until it fulfils the criteria described by its termination function, accordingly, returns four variables as shown in Appendix 4:

(1) The best population found during the GA→`x`.
(2) The final population →`endPop`.
(3) The best solution tracked over generations →`bPop`.
(4) The best value and average value for each generation →`traceInfo`.

The best population can be plotted to give an insight into how the genetic algorithm converged to its final values as illustrated in Fig.5. This figure is obtained assuming an arbitrary transfer function (not necessary that of our power system) just to show the PID variation with population. Actually for our system and for the proposed **"MSE as well as ISE"** criterion one gets similar good results which are shown in details in the next section. Fig. 6 shows the corresponding step response. This figure also shows the superiority of the PID controller tuned with GA over the response if only Ziegler-Nicholes is implemented for the PID controller design.

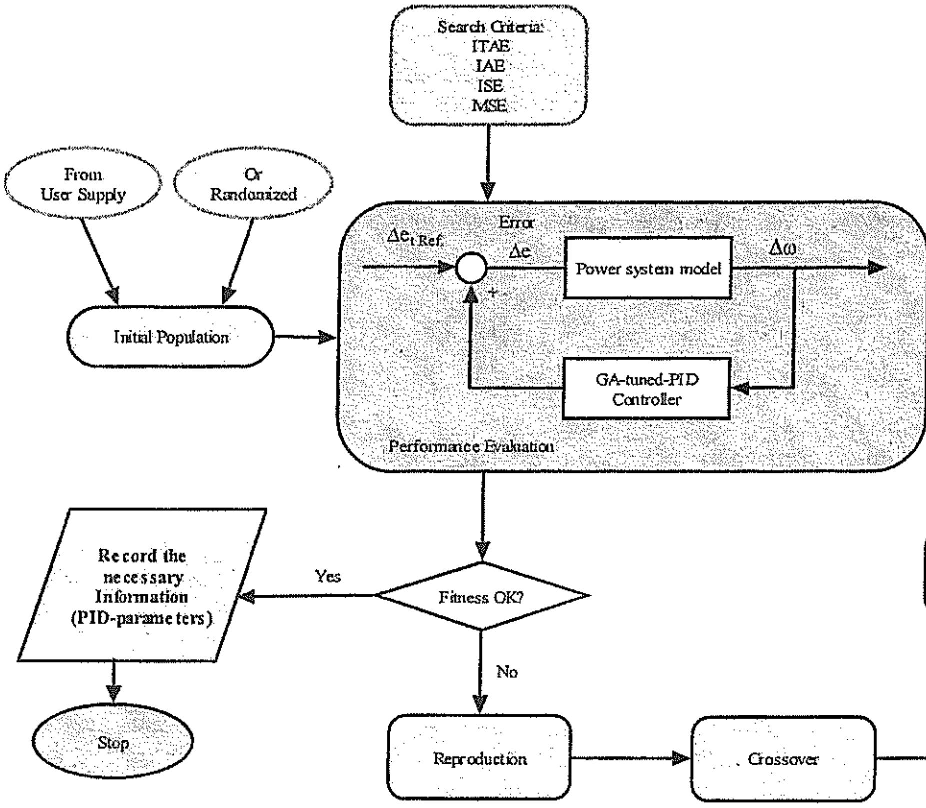

Fig. 4. Block-Diagram of the GA Processes

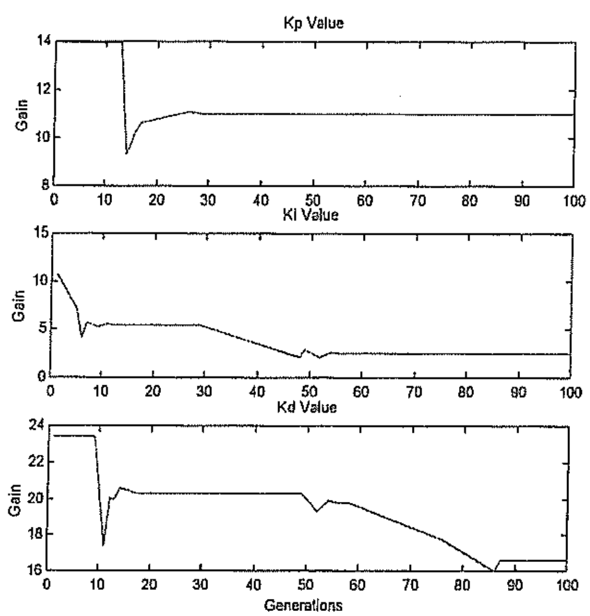

Fig. 5. Convergence of the best population to the final values of PID for a test system

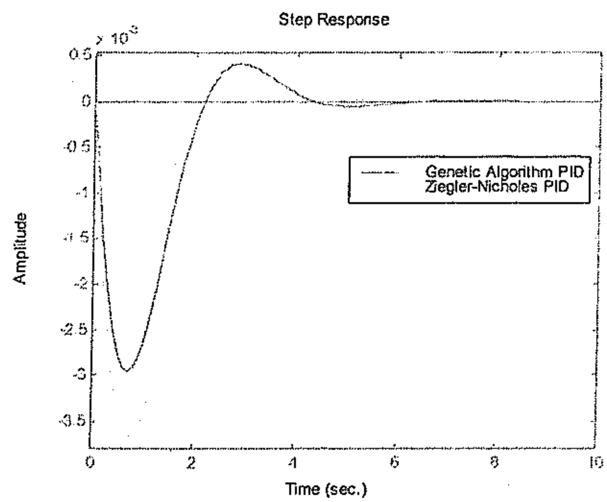

Fig. 6. Step response for the controlled power system comparing the response of GA tuned PID and PID controller just designed with Ziegler-Nichols

3.3. Simulation and Assessment of the Four Search Criterion

In order to evaluate which of the previously mentioned four performance criterion produce the best results when used in conjunction with a genetic algorithm, an objective function was created for each individual performance criterion as given in

Appendix 3 with the algorithms given in Appendix 7. The same Genetic Algorithm was used for each objective function. In the Matlab m-Code, the genetic algorithm was initialized with a population of "thirty" and was iterated for 200 generations. The total number of mutations was set to "three" and each of the bounds was set to lie between +10 and +110 as seen in Appendix 6. All of the genetic algorithms had the exact same initial conditions. The Matlab command "rand('state',0);" was used. This command guarantees that each population is initialized to the same set of values. Fig. 9 compares the step response of the Ziegler-Nicholes designed PID controller Appendix 5 versus a Genetic Algorithm tuned PID controller using each of the objective functions. Table 1 describes the steady state characteristics of each of the controlled systems. For the above mentioned conditions, it can be seen that the ISE and MSE objective functions perform almost identically, having a smaller rise time, smaller overshoot and shorter settling time than the other controllers. Each of the genetic algorithm-tuned PID controllers outperforms the Ziegler-Nichols tuned controller in terms of rise time and overshoot but only the ITAE and IAE objective functions outperform it in terms of settling time. Accordingly, the results recommend that either the MSE or ISE objective function should be chosen as the primary performance criterion due to its smaller rise time, shorter settling time, and smaller overshoot than any other method. Also, these two methods are advantageous in conjunction with a slightly faster compile time due to there being just one multiplication to be carried after the error has been calculated. This, partially, ensures the fact that MSE has been considered as an efficient measure of control and quality for many years. This leads to the conclusion that either MSE or ISE are ideal search criteria for tuning PID controller of the power system under study.

4. CONCLUSIONS

This paper has presented a genetically tuned PID controller for actual models of power systems having uncertain parameters to replace the conventional power system stabilizer which is efficient only for certain parameters and linear models. Four different search criteria are implemented to ensure the robustness of the proposed controller under parameter uncertainties. These include, Integral of Time multiplied by Absolute Error (ITAE), Integral of Absolute Magnitude of the Error (IAE), Integral of the Square of the Error (ISE), and Mean of the Square of the Error (MSE). In order to evaluate which of these performance criteria produce the best results when used in conjunction with a genetic algorithm, an objective function was created for each individual performance. Simulation results showed that either the MSE or ISE objective function should be chosen as the primary performance criterion. Transient response associated with these two objective functions are characterized with small rise time, short settling time, and small overshoot.

5. ACKNOWLEDGMENT

The author gratefully acknowledges the patience of his wife Sylvia Kromberg during the language review and text typing.

6. REFERENCES

[1] deMello, F.P. and Concordia, C. "Concepts of synchronous machine stability as affected by excitation control", IEEE Trans. Vol. PAS-87, Pp.316-329, 1969.

[2] Larsen, E.V. and Swann, D. A., "Applying power system stabilizer", Part I, II, III, IEEE Trans., PAS-100, Issue 6 , Pp. 3017-3046, 1981.

[3] Concordia, C., "Power System Objectives, Side Effects: Good and Bad", IEEE Power Engineering Review, September 1990.

[4] Yang, T.C., Munro, N. and Brameller, A. , "Application of optimal regulator design method through the inverse problem approach to the power system stabilizer design", Journal of Electric Power & Energy Conversion, Vol. 13, No.1 , Pp.57-62, Feb. 1991.

[5] Yang, T.C. and Munro, N., "Power system stabilizer design based on the pole assignment technique for SIMO systems", Journal of Electric Power & Energy Conversion, Vol. 13, No.6, Pp.298-302, Jan. 1992.

[6] Yang, T.C.,"Synchronous generator stabilizer design through incomplete state feedback", Journal of Electric Power & Energy Conversion, Vol. 16, No.2, Pp.91-95, Sept. 1994.

[7] Wang, L, "Damping effects of supplementary excitation control signals on stabilizing generator oscillations", Journal of Electric Power & Energy Conversion, Vol. 18, No.1, Pp.47-53, 1996.

[8] Baskar, S.; Subbaraj, P.; "Design of SSMFs fuzzy logic based power system stabiliser using genetic algorithms", TENCON '98 (1998 IEEE Region 10 International Conference on Global Connectivity in Energy, Computer, Communication and Control), Pp. 548-551, Volume 2, 17-19 Dec. 1998.

[9] Abdel-Hamid , A, M.; "A Novel Approach To Improve The Performance Of Fuzzy-Logic Based Controllers For Systems With Wide Range Disturbances And Uncertain Parameters", Engineering Research Bulletin, Faculty of engineering, Minufiya university, Shebin El- Kom, Egypt. Vol.22, No.3, Pp. 41-56, October 1999.

[10] Wu Chi-Jui; Hsu Yuan-Yih; "Design of self-tuning PID power system stabilizer for multimachine power systems", IEEE Transactions on Power

Systems, Volume 3, Issue 3, Pp. 1059-1064, Aug. 1988.

[11] Hjdfhjs Linkens, D.A., & H.O. Nyongesa, 'Genetic algorithms for fuzzy control', IEE Proc. Control Theory Appl., Vol. 142, No. 3, pp.161-185, 1995.

[12] Chian-Chuang Ding; King-Tan Lee; Chee-Ming Tsai; Tsong-Liang Huang; "Optimal design for power system dynamic stabilizer by grey prediction PID control", IEEE International Conference on Industrial Technology, 2002, IEEE ICIT '02, Volume 1, Pp. 279-284, 11-14 Dec. 2002.

[13] Dobrescu, M.; Kamwa, I.;" A new fuzzy logic power system stabilizer performances", IEEE PES, Vol. 2, Pp. 1056-1061, 10-13 Oct. 2004.

[14] Kristiansson, B. and Lennartson, B, "Robust Tuning of PI and PID controller," CONTROL SYSTEMS MAGAZINE, pp. 55-69, FEBRUARY 2006.

[15] Mishra, S.;" Neural-network-based adaptive UPFC for improving transient stability performance of power system", IEEE Transactions on Neural Networks, Volume 17, Issue 2, Pp.461-470, March 2006.

[16] Hakim, Ermanu A.; Suprijanto, Adi; Maurdhi, Heri P.;" PSS based on optimal fuzzy PID with Particle Swarm Optimization", IEEE- IPEC 2007, Pp.1396-1400, 3-6 Dec. 2007.

[17] Kundur, Prabha;; "Power System Stability And Control", McGraw-Hill, 1994.

[18] Abdel-Hamid, A.M. and H.A. Nour Eldin, "A Modal Analysis Based Load Frequency - Control Considering Parameter Uncertainties And Generation Rate Constraints", The 42nd Internationales Wissenschaftliches Kolloquim, Band.3, pp. 705-710, Technische Universitaet Ilmenau, Germany, sept.1997.

[19] O' Mahony, T., Downing,C.J. and Klaudiuz, F., "Genetic Algorithms for PID Parameter Optimisation: Minimising Error Criteria",[online], URL: http://www.pwr.wroc.pl/~i-8zas/kf_glas00.pdf

7. APPENDICES

7.1. Appendix 1. Nomenclature & System Data

M	*Machine inertia coefficient =10 s.*
D	*Machine damping coefficient =0*
ω_b	*System base angle frequency = 377.*
T_M	*Machine mechanical torque*
T_E	*Machine electrical torque*
ω	*Machine speed*
δ	*Angle between machine quadrature axis and infinite bus*
T'_{do}	*DIRECT axis transient open circuit constant =6.0 s.*
E_{fd}	*Generator field voltage*
$e_{t\,Ref.}$	*Terminal reference voltage*
e_{PSS}	*PSS output voltage signal*
e_t	*Voltage error signal*
E'_q	*q-axis component of voltage behind transient reactance*
K_a	*Exciter amplifier gain = 25.0*
Ta	*Exciter amplifier time constant =0.05 s.*
Xd	*d-axis reactance =1.6 p.u.*
Xq	*q-axis reactance =1.55 p.u.*
$X'd$	*d-axis transient reactance =0.32 p.u.*
P	*Machine active power loading =1. p.u.*
Q	*Machine reactive power loading =0.2 p.u.*
Xe	*transmission line reactance =0.4 p.u.*
Δ	*Prefix, stands for small change.*

7.2. Appendix 2. K-Constants Calculations

The fourth order model of synchronous machine is described by these equations:

$$\Delta T_m - \Delta P = M\frac{d^2\Delta\delta}{dt^2}$$

$$\Delta P = K_1\Delta\delta + K_2\Delta E_q^{'}$$

$$\Delta E'_q = \frac{K_3}{1+sT_{do}^{'}K_3}\Delta E_{fd} - \frac{K_3K_4}{1+sT_{do}^{'}K_3}\Delta\delta$$

$$\Delta V_t = K_5\Delta\delta + K_6\Delta E_q^{'}$$

For a steady-state operating point P_0, Q_0 and V_{t0}, one can calculate the initial conditions and K-Constants as follows:

$$i_{q0} = \frac{P_0V_{to}}{\sqrt{(P_0x_q)^2+(V_{t0}^2+Q_0x_q)^2}}$$

$$v_{d0} = i_{q0}x_q$$

$$v_{qo} = \sqrt{V_{t0}^2 - v_{t0}^2}$$

$$i_{d0} = \frac{Q_0 + x_q i_{q0}^2}{v_{q0}}$$

$$E_{q0} = v_{q0} + i_{d0}x_q$$

$$E_0 = \sqrt{(v_{d0}+x_ei_{q0})^2 + (v_{q0}-x_ei_{d0})^2}$$

$$\delta_0 = \tan^{-1}\frac{(v_{d0}+x_ei_{q0})}{(v_{q0}-x_ei_{d0})}$$

$$K_1 = \frac{x_q - x_d^{'}}{x_e + x_d^{'}}i_{q0}E_0\sin\delta_0 + \frac{E_{q0}E_0\cos\delta_0}{x_e+x_q}$$

$$K_2 = \frac{E_0\sin\delta_0}{x_e + x_d^{'}}$$

$$K_3 = \frac{x_d^{'} + x_e}{x_d + x_e}$$

$$K_4 = \frac{x_q - x_d^{'}}{x_e + x_d^{'}}E_0\sin\delta_0$$

$$K_5 = \frac{x_q}{x_e + x_q}\frac{v_{d0}}{V_{t0}}E_0\cos\delta_0 - \frac{xd'}{x_e + x_d^{'}}\frac{v_{q0}}{V_{t0}}E_0\sin\delta_0$$

$$K_6 = \frac{x_e}{x_e + x_d'} \frac{v_{q0}}{V_{t0}}$$

7.3. Appendix 3. Performance Criteria

In this paper four different objective functions are defined. They are written in the m-code based on error performance criterion. All these objective function are fundamentally the same except for the section of code that defines the specific error performance criterion being implemented. A summary of these criteria can be given as:

(1) $I_{MSE} = \frac{1}{n} \sum_{i=1}^{n} (e(t))^2$

(2) $I_{ISE} = \int_0^T e^2(t)dt$

(3) $I_{IAE} = \int_0^T |e(t)| dt$

(4) $I_{ITAE} = \int_0^T t|e(t)| dt$

7.4. Appendix 4. Results of the necessary steps required to find out a solution by GA

The basic GA algorithm involves the generation of a population of possible solutions, evaluation of the solutions according to a fitness function, selection of a set of fit "parent" solutions, and finally reproduction of those parents to generate a new population of possible solution. Matlab results corresponding to these steps are:

```
» x
x =
   65.8546    15.2257   109.7120     0.9993
» endPop
endPop =
   65.8547    15.2257   109.7120     0.9993
   65.8547    15.2257   109.7120     0.9993
   65.8546    15.2257   109.7120     0.9993
   65.8547    15.2257   109.7120     0.9993
   65.8547    15.2257   109.7120     0.9993
   65.8546    15.2257   109.7120     0.9993
   65.8546    15.2257   109.7120     0.9993
   65.8546    15.2257   109.7120     0.9993
   65.8546    15.2257   109.7120     0.9993
   65.8546    15.2257   109.7120     0.9993
   65.8546    15.2257   109.7120     0.9993
   65.8546    15.2257   109.7120     0.9993
   65.8546    15.2257   109.7120     0.9993
   65.8546    15.2257   109.7120     0.9993
   65.8546    15.2257   109.7120     0.9993
   65.8546    15.2257   109.7120     0.9993
   65.8546    15.2257   109.7120     0.9993
   65.8546    15.2257   109.7120     0.9993
   65.8546    15.2257   109.7120     0.9993
   65.8546    15.2257   109.7120     0.9993
» bPop
bPop =
    1.0000    63.3799    11.2827   108.8196     0.9993
   19.0000    61.2791    17.4797   109.5244     0.9993
   58.0000    66.7072    15.3340   109.7071     0.9993
  100.0000    65.8546    15.2257   109.7120     0.9993
» traceInfo
traceInfo =
    1.0000     0.9993     0.9991     0.0001
    2.0000     0.9993     0.9992     0.0001
    3.0000     0.9993     0.9992     0.0001
    4.0000     0.9993     0.9992     0.0000
    5.0000     0.9993     0.9992     0.0000
```

7.5. Appendix 5. m-Code for Ziegler-Nichoes PID calculations

```
%-----------------------------------------------------------
% A Program to design the PID parameters.
% Calculations are based on Ziegler-Nicholes.
% Written by A. M. Abdel-Hamid.
% Last modified: April 2008.
%-----------------------------------------------------------
% The T.F. Of Power System:
den1=[1 20.463 80.95 841.166 1771.779]
num1=[-11.121]
sysr1=tf(num1,den1)
rlocus(num1,den1);sgrid
% Crossover frequency is jw=1*i; i.e. w=1.0
[k,poles] = rlocfind(num1,den1,1*i)
wc=1
Tc=2.*3.14/wc
% The system gain Kc=k:
Kc=k
% The Ziegler - Nicholes calculations to find out
% Kp,Ki, and Kd as explained in  the given tables
TI=Tc/2
TD=Tc/8
Kp=0.6*Kc
Ki=Kp/TI
Kd=Kp*TD
% These lines are just to make sure that
% the open loop state space form give the
% same open loop eigenvalues.
[A,B,C,D] = tf2ss(num1,den1)
eig(A)
roots(den1)
```

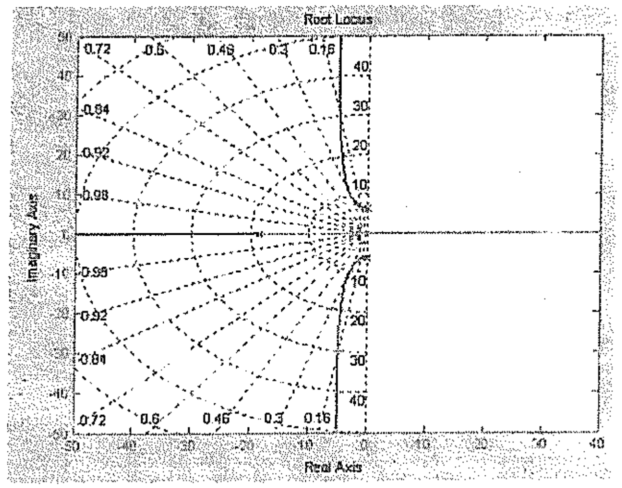

Fig.7: Root locus plot of power system model [17]:

$$G(s) = \frac{-11.121}{s^4 + 20.463s^3 + 80.95s^2 + 841.166s + 1771.779}$$

7.6. Appendix 6. m-Code for GA initialization

```
%-----------------------------------------------------------
%Defining sysr1
den1=[1 20.463 80.95 841.166 1771.779]
num1=[-11.121 0]
sysr1=tf(num1,den1);
%-----------------------------------------------------------
%Initializing the genetic algorithm
populationSize=30;
variableBounds=[10 110;10 110;10 110];
evalFN='PID_objfun_ITAE_Hefzon';
%Change this to relevant object function
evalOps=[];
options=[1e-6 1];
initPop=initializega(populationSize,variableBounds,evalFN,evalOps,options);
%-----------------------------------------------------------
%Setting the parameters for the genetic algorithm
bounds=[10 110;10 110;10 110];
evalFN='PID_objfun_ITAE_Hefzon';%change this to relevant object function
evalOps=[];
startPop=initPop;
opts=[1e-6 1 0];
termFN='maxGenTerm';
termOps=200;
selectFN='normGeomSelect';
selectOps=0.08;
xOverFNs='arithXover';
xOverOps=4;
mutFNs='multiNonUnifMutation';
mutOps=[8 200 3];
%mutFNs='unifMutation';
%mutOps=8;
```

7.7. Appendix 7. Algorithm of the different search criteria

Performance Criteria	Symbol	Mathematical description and m-code

Mean of the square of the Error	MSE	***Equation:***
		$I_{MSE} = \frac{1}{n} \sum_{i=1}^{n} (e(t))^2$
		m-code:
		`%Calculating the error`
		`for i=1:301`
		`error(i) = 1-y(i);`
		`end`
		`%Calculating the MSE`
		`error_sq = error*error';`
		`MSE=error_sq/max(size(error));`

Integral of the Square of the Error	ISE	***Equation:***
		$I_{ISE} = \int_0^T e^2(t)dt$
		m-code:
		`%Calculating the error`
		`for i=1:301`
		`error(i) = 1-y(i);`
		`end`
		`%Integral of Squared Error`
		`error=error*error';`
		`ISE=sum(error);`

Integral of Absolute Magnitude of the Error	IAE	***Equation:***
		$I_{IAE} = \int_0^T \lvert e(t) \rvert dt$
		m-code:
		`%Calculating the error`
		`for i=1:301`
		`error(i) = 1-y(i);`
		`end`
		`%Integral of absolute error`
		`IAE=sum(abs(error));`

Integral of Time multiplied by Absolute Error	ITAE	***Equation:***
		$I_{ITAE} = \int_0^T t \lvert e(t) \rvert dt$
		m-code:
		`%Calculating the error`
		`for i=1:301`
		`error(i) = (abs(1-y(i)))*t(i);`
		`end`
		`%Integral of Time multiplied`
		`% by Absolute Error`
		`ITAE=sum(error);`

7.8. Appendix 8. Brief Explanation of the Important GA Processes

It is clear from Fig.4 that during GA some important simulations should be executed. These can be summarized as:

[1] Representation and Fitness:

The first step in creating a GA is to select a solution representation and a fitness function. The solution representation is, usually, a fixed-length string of units (bits, real numbers, letters, etc.), and this is still the standard representation until now. Each string must represent a possible solution in some non-arbitrary way. The fitness function is the essence of the problem: it provides the means by which the quality of a solution may be assessed, and the probability that a solution will reproduce.. If we are using a GA to find the tuned PID parameters, a solution which finds ***Kp***, ***Ki***, and ***Kd*** within industrial limits ought to be fitter than a solution which finds negative or too large PID parameters.

[2] Generation of an initial population:

The initial population is typically generated at random; such that each string represents a potential solution (often impossible solutions are excluded). Alternatively، the population may be seeded in areas where it is likely to find a solution, potentially shortening the time required to solve the problem

[3] Selection of solutions:

A fitness-based selection method is used to choose those solutions which will produce the next generation. The selection method is biased towards individuals of higher fitness, in order that better genetic material can persist in the population, and be improved upon through reproduction. There is several different selection schemes used in GAs. One common method, *fitness proportionate selection*, selects parents with a probability which is directly proportional to their fitness. This requires evaluating the fitness of every solution in the population. A second method which may require fewer fitness evaluations is *tournament selection*. In this method, solutions are randomly selected to participate in a "*tournament*"; the solution with the highest fitness is selected, and the process repeats until enough parents are chosen. Most selection methods are stochastic, and so may allow a small number of less-fit solutions to reproduce.

[4] Reproduction:

Reproduction, generally, consists of two parts, crossover and mutation:

I. Crossover:

Crossover is the basic method of recombining genetic material from two parents. Crossover commonly involves randomly selecting some number of crossover points, and exchanging those alleles which lie between the points. For example، if the two parents below (binary coded population) undergo two-point crossover at the positions indicated with " v"، they may produce either of the two "children" shown in Fig.8:

Parent 1: 0 0 0 v 0 0 0 0 v 0

Parent 2: 1 1 1 v 1 1 1 1 v 1

Child 1: 0 0 0 | 1 1 1 1 | 0

Child 2: 1 1 1 | 0 0 0 0 | 1

Fig.8: Crossover and Mutation processes over a bit coded representation.

II. *II. Mutation:*

Mutation is the disruption (interference) of genetic material after crossover. By introducing random variations into the child population, we can ensure that the diversity of the population remains large. Mutation generally depends on a mutation rate, which is the probability that any one allele will mutate to a new value. A typical mutation on one allele may involve ***"flipping" one bit from 0 to 1*** or visa versa in a ***binary string***, or ***adding a random value*** to an allele in a string of ***real numbers***.

Table 1 Controller parameters and Performance of GA search criteria compared to ZN

	MSE	*ISE*	*IAE*	*ITAE*	*ZN*
P	107.8095	108.6495	68.8188	10.0086	101.4504
I	109.9956	109.9972	109.9992	50.3717	32.3090
D	84.9798	61.6678	109.9827	109.9676	79.6385
% Overshoot	4.4054	4.6504	31.0345	31.1237	4.6342
Tr(Rise time)	0.3924	0.2226	0.5007	0.4385	0.1082
Ts(settling time)	4.4087	4.2056	9.4305	4.6243	5.6504

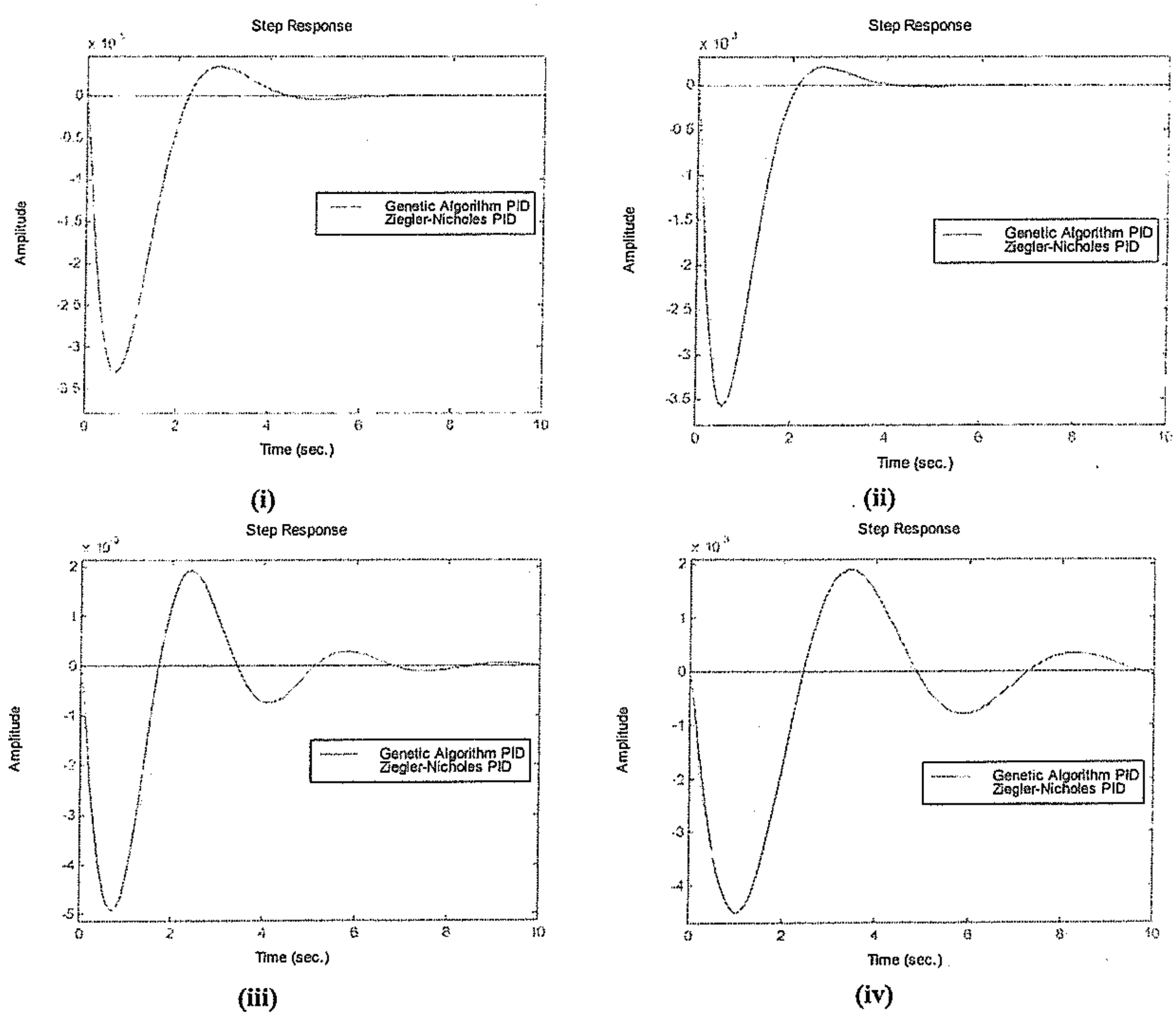

Fig. 9 GA tuned PID Controller (red) compared with Ziegler-Nichols designed PID controller (green-dashed) using **(i)** MSE, **(ii)** ISE, **(iii)** IAE and **(iv)** ITAE as performance criterion.