Load Frequency Control: A generalized Neural Network Approach

D. K. Chaturvedi, Dayalbagh Educational Institute
P. S. Satsangi, Dayalbagh Educational Institute
P. K. Kalra

Available at: https://works.bepress.com/dk_chaturvedi/5/
Load frequency control: a generalised neural network approach

D.K. Chaturvedi*1, P.S. Satsangi2, P.K. Kalra3

1Dayalbagh Educational Institute, Dayalbagh, Agra (U.P.), 282 005, India
2Indian Institute of Technology, Kanpur (U.P.), 208 016, India

Received 25 May 1998; received in revised form 29 October 1998; accepted 28 January 1999

Abstract

Variation in load frequency is an index for normal operation of power systems. When load perturbation takes place anywhere in any area of the system, it will affect the frequency at other areas also. To control load frequency of power systems various controllers are used in different areas, but due to non-linearities in the system components and alternators, these controllers cannot control the frequency quickly and efficiently. Simple neural networks which are in common use at present have various drawbacks like large training time, requirement of large number of neurons, etc.

The present work deals with the development of a non-linear neural network controller using a generalised neural network. The drawbacks of existing neural networks have been overcome in the generalised neuron structure which has been developed to control the deviations in load frequency of a power system. © 1999 Elsevier Science Ltd. All rights reserved.

Keywords: Load frequency control; Neural network approach

Nomenclature

df

dp pref

dph

dpt

dpe

ddf, dpdt, dpdph

dt

Kg

Kb, Kg, Kp

Kg

Tr, Td, Tp

R

dp12

Subscripts 1, 2

1. Introduction

A large number of controllers are used to maintain a power system in a normal state of operation. As demand fluctuates from its normal operating value the state of the system changes. To maintain the system at a normal operating state different types of controllers based on classical linear control theory have been developed in the past [2,10,12,14,25]. Most load frequency controllers are primarily composed of an integral controller. The integrator gain is set to a level that compromises between fast transient recovery and low overshoot in the dynamic response of the overall system [14,17,22,23,26]. This type of controller is slow and does not allow the designer to take into account possible non-linearities in the generator unit.

The inherent non-linearities in system components and synchronous machines have led researchers to consider neural network techniques to build a non-linear ANN controller with high efficiency of performance [5,11,13].

The existing simple neural networks have numerous deficiencies as stated below:

1. The number of neurons required in hidden layers is large for complex function approximation.
2. The number of hidden layers required for complicated functions may be greater than three. Though it has been reported that a network with only three layers can approximate any functional relation, it is found that the
3. The fault tolerant capabilities of the existing neural networks are very limited.
4. Existing neural networks require a large number of unknowns to be determined for complex function approximation. This increases the requirement of the minimum number of input–output pairs.
5. The training time required is dependent to a large extent on the type of input–output mapping chosen (like ∆X–Y, X–Y, etc.) [7].
6. The training time is also dependent on the normalisation ranges used for the input and output data.

In the present work an automatic load frequency controller has been developed using Neural Networks to regulate the power output and system frequency by controlling the speed of the generator with the help of water or steam flow control. The aim of this controller is to restore the frequency to its nominal value in the shortest time whenever there is a change in the demand. The action of the controller should be coupled with minimum frequency transients and zero steady-state error.

The following ideas were formulated to overcome the deficiencies of the existing neural networks:

1. In Fuzzy Logic Systems, the compensatory aggregation operators perform better than the individual min and max operators. This is because an appropriate weighted combination of these two is used in the compensatory operator [5]. In a similar way the Sum and Product neurons could be combined to develop a Generalised neuron which may be expected to perform better than the individual neurons.
2. Development of a generalised neuron model which exhibits characteristics of all types of existing neurons. Such neuron models will be general enough to accommodate properties of simple as well as higher order neurons.
3. The neuron model should be flexible enough to accommodate variations in mappings and hence drastically reduce the total number of neurons in the neural network.
4. The generalised neuron model should reduce the requirement of the total number of hidden layers. This would result in a neural network model which is computationally efficient.
5. The smaller number of neurons needed to model any function in the generalised neural network should reduce the number of free parameters associated with the neurons, thus reducing the amount of data required for training.

The above ideas have been incorporated in the generalised neuron structure developed. This neuron model has been used to develop a Generalised Neural Network (GNN) which is used in the development of a load frequency controller.

Conventional integral controllers base their action on the change in the frequency which makes their response more delayed than that of a ANN controller whose action is based on the rate of change of frequency. The GNN load frequency controller developed also makes use of the rate of change of frequency to estimate the electric load perturbation. The load perturbation estimate could be obtained either by a linear estimator or by a non-linear ANN estimator.

The GNN has been used to develop load frequency controllers for use in single area systems and in multi-area systems. The performance of the GNN load frequency controller has been compared with that of conventional integral controller as well as the ANN controller.
2. Modelling and simulation of plant with conventional integral controller

2.1. Single area system

In the generating plant, mechanical power is produced by a steam/hydro turbine and delivered to synchronous generator, which ultimately generates the electrical power and supplies it to the loads. The quality of supply depends on the voltage and frequency variations. The frequency and the voltage at the generator terminals are determined by the turbine steam flow. It is also affected by the change in load/demand. If the load on generator terminals suddenly increases or decreases, the generator shaft slows down or accelerates, which finally decreases or increases the frequency of the supply [1, 4, 13, 20]. To control these variations in voltage and frequency, sophisticated controllers are required, which immediately detect the load variation and command the steam admission valve to open or close to counter balance the load variation. In conventional control systems, an integral controller controls the frequency variation to zero as early as possible by getting optimum value of the integral gain \( K_i \).

In power systems most of the devices are extremely non-linear; one usually likes to linearize the plant and to think of different variables in terms of their fluctuations about a given operating point. Non-linearities are then modelled by making the parameters of linearized system functions of the operating point. The resulting small signal models consist of linear operators having variable parameters whose values depend upon the state of the system. The last step in modelling consists of replacing all small signals by their Laplace transform and to represent the linearized devices by transfer functions. A Laplace transform domain small signal model of the single area system is given in Fig. 1. The global control action can be obtained by adding integral control action and the control signal obtained from the speed regulator. This global control action produces the hydraulic amplifier output \( 'dph' \), The turbine block is a first-order transfer function, whose input is \( 'dph' \) and the output is turbine mechanical power \( 'dpt' \). The turbine output and the electrical load perturbation given in the generator block, gives \( 'dl' \) as the output. To simulate the single area system using MATLAB, DYNAMO equations are required to be in discrete time domain. The DYNAMO equations are evaluated repeatedly to generate a sequence of points equally spaced in time.

These equations consisting of level equations, rate equations, auxiliary equations and exogenous variables and constants [15, 16]. The DYNAMO equation for single area system is mentioned in the following.

2.1.1. Level equations

These equations are used for the calculation of present value of level (state) variable from the previous value of it [15, 16].

\[
\begin{align*}
\text{df} &= df + dt*\text{ddf} \\
\text{dpt} &= dpt + dt*\text{dpt} \\
\text{dph} &= dph + dt*\text{dph}
\end{align*}
\]

\[ (1) \]

\[ (2) \]

\[ (3) \]
2.1.2. Rate equations

The rate equations indicate the rate of change of level equations and these rate variables remain constant for the sample time of simulation (dt).

\[ ddf = (K_p \cdot dpt - K_p \cdot dpe - df)/T_p \]  \hspace{1cm} (4)

\[ ddpt = (K_i \cdot dph - dpt)/T_i \]  \hspace{1cm} (5)

\[ ddph = (K_h \cdot dpref - K_h \cdot dph/R - dph)/T_h \]  \hspace{1cm} (6)

dpref = dpref - K_i \cdot df

2.1.3. Auxiliary equations

In addition to the level and rate equations some auxiliary equations, system parametric and initial value equations are also used to facilitate the calculations of level and rate variables of the system. The auxiliary equations add convenience and clarity to complex system models [15,16]. The auxiliary equation for single area system is given below:

\[ \text{dpref} = \text{d pref} - K_i \cdot df \]  \hspace{1cm} (7)
2.1.4. System parameters and constants

The gain and time constants of the turbine, hydraulic amplifier and generator are as follows:

\[ K_p = 1.0, \quad T_p = 1.0, \quad K_w = 120 \text{ Hz/pu/MW}^2, \]
\[ T_a = 80 \text{ ms}, \quad T = 0.3 \text{ s}, \quad T_d = 20 \text{ s}, \]
\[ R = 2.4 \text{ Hz/pu/MW}, \quad K_i = \text{ any non-negative value.} \]

While simulating the aforementioned model, it was found that for any non-negative value of integral gain except zero and for any step perturbation the system is stable (i.e., \( df \) converges to zero). The final values of the level variables are \( df = 0.0, \) \( dp = dpe, \) and \( dp = dpe/K_i. \)

Fig. 2 shows the frequency variations of single area system after simulating the above mentioned model for different values of integrator gain \( K_i \) for 10% step disturbance in load. The critical value of \( K_i \) can be determined from the formula

\[ K_{critical} > \frac{1}{4T_a^2R} \left( 1 + \frac{K_p}{R} \right)^2. \]

The change in turbine power and change in reference power are also shown in Figs. 3 and 4.

2.2. Two area system

The two area system consists of two single area systems connected through a power line called tie line: each area feeds its user pool and the tie lines allow electric power to flow between the areas. Because both areas are tied together, change in the demand of one area affects the frequencies of both areas. The DYNAMO equations of two area system have been developed as follows, for simulating system for different values of perturbations.

- Level equations
  \[ df_1 = df_1 + dt^*df_1 \]
  \[ df_2 = df_2 + dt^*df_2 \]
  \[ dp12 = dp12 + dt^*dp12 \]

- Rate equations
  \[ ddf_1 = (K_p^*dpref_1 - (K_p/(R_1 + 1))^*df_1 - K_p^*dpe_1)/T_p \]
  \[ ddf_2 = (K_p^*dpref_2 - (K_p/(R_2 + 1))^*df_2 - K_p^*dpe_2)/T_p \]
  \[ dp12 = (2^*\pi^*T_p^*)*(df_1 - df_2) \]

- Auxiliary equations
  \[ dpref_1 = dpref_1 - K_i^*df_1 - dp12^*K_i \]
  \[ dpref_2 = dpref_2 - K_i^*df_2 + dp12^*K_i \]

- System parameters and constants
  \[ K_p = 120 \text{ Hz/pu MW}, \quad K_w = 120 \text{ Hz/pu MW} \]
  \[ T_p = 20 \text{ s}, \quad T_d = 20 \text{ s} \]
  \[ R_1 = 2.4 \text{ Hz/pu MW}, \quad R_2 = 2.4 \text{ Hz/pu MW}. \]
The above developed DYNAMO model has been simulated for 10% step disturbance in area 2 with and without integral controller of gain $K_i = 0.05$ and the results obtained have been shown in Figs. 5 and 6 for frequency variations for $R = 2.4$ and 6, respectively. It was found that for different values of $R$ the steady state frequencies are different. It is also seen that the higher value of $R$ creates more ringing in the system.

From the above simulation results it is found that for any non-negative integrator gains, the frequency as well as tie line power transients die out and the steady state value reaches zero.

3. Generalised neural network controller

In the earlier section, the plant models used so far were linearized models. It was assumed that the operating point of the plant did not change much when a step load perturbation occurred on the bus, and that, therefore, all the plant parameters could be kept constant. In practice though, the constant characterising the speed regulator $R$ depends in a highly non-linear manner upon the turbine power [8,9]. The presence of this non-linearity and the slowness of traditional integral controllers encouraged us to develop Neural Network Controller. The ANN controller using two layered feed-forward neural network has been developed by Beanfays in 1994 [3] for single area as well as two area systems. Djukanovic [11] also developed an optimal load frequency controller using ANN and Fuzzy set theoretical approach. As mentioned above that the existing simple neural network has various limitations like huge training data, large training time required, large number of neurons and two or more than two layers required to map the complicated functions. To overcome these drawbacks, Generalised Neural Network (GNN) has been developed by choosing different compensatory aggregation functions and threshold functions in the same neuron. The backpropagation-through-time learning algorithm [18,19,24] is used, which is an extension of the well-known back propagation algorithm [21,24]. The feed-forward neural network is static in nature. It means there is no time dependency existing between input and desired outputs. The load frequency control problem is dynamic and to control such a problem, there is a genuine need of Generalised Neural Network which can quickly handle all type of situations and control the plant dynamics.

3.1. Generalised neuron modelling and its training

In the generalised neuron model both summations as well as product have been taken as the aggregation functions and the output of these aggregation functions have been passed through the sigmoid and gaussian functions, respectively. Finally, the outputs are summed up to get the neuron output. The neuron structure has been shown in Fig. 7. The output
of the neuron can be mathematically written as
\[ O_i = W_{2i}O_2 + (1 - W_{2i})O_1. \]

The above mentioned neuron model is known as summation type compensatory neurons model, since the output of the sigmoidal and gaussian functions have been added up.

### 3.2. Learning algorithm of generalised neuron model

The following steps are involved in the training of Generalised Neural Network [6]

**Step 1** The output of \( \Sigma \)-part of generalised neuron is
\[ O_{2i} = f_2(W_{2i}x_2 + x_{2a1}). \]

**Step 2** The output of \( \Pi \)-part of generalised neuron is
\[ O_{1i} = f_1(W_{1i}x_1 + x_{1a1}). \]

**Step 3** The output of Generalised Neuron can be written as
\[ O_i = W_{2i}O_2 + (1 - W_{2i})O_1. \]

**Step 4** After calculating the output of Generalised Neuron in the forward pass of feedforward backpropagation neural networks, it is compared with the desired output to find the error and then it is minimised to train Generalised Neural Network (GNN). Hence in this step the output of GNN is to be compared with the desired output to get error for the \( i \)th set of input

\[ Error_{E_i} = (Y_i - O_i). \]

Then it is necessary to calculate the sum squared error for convergence while training.

\[ E_p = 0.5 \sum E_i^2. \]

A multiplication factor of 0.5 has been taken for simplifying the calculations.

**Step 5** Reverse pass for modifying the connection strength.

(a) Weights associated with \( \Sigma \) and \( \Pi \)-parts of Generalised Neuron
\[ W_{2i}(k) = W_{2i}(k - 1) + \Delta W_{2i} \]
where
\[ \Delta W_{2i} = \eta \delta_{2i}(O_2 - O_{1i}) + \alpha W_{2i}(k - 1) \]
and
\[ \delta_{2i} = (Y_i - O_i) \]

(b) Weight associated with the inputs of \( \Sigma \)-part of Generalised Neuron
\[ W_{2i}(k) = W_{2i}(k - 1) + \Delta W_{2i} \]
where
\[ \Delta W_{2i} = \eta \delta_{2i}x_i + \alpha W_{2i}(k - 1) \]
and
\[ \delta_{2i} = \sum \delta_{2}(1 - O_2) \]

(c) Weight associated with the input of \( \Pi \)-part of Generalised Neuron
\[ W_{1i}(k) = W_{1i}(k - 1) + \Delta W_{1i} \]
where
\[ \Delta W_{1i} = \eta \delta_{1i}x_i + \alpha W_{1i}(k - 1) \]
and
\[ \delta_{1i} = \sum \delta_{1}(1 - W_2X_2) \]
\[ \text{net}_{1i} = (1W_1X_1 + X_{a1}) \]

3.3. **GNN controller of single area system**

It has been shown above that the level variable vector \([df df dph]\) of a single area system controlled by an integral controller eventually converged to a steady-state value equal to \([0 df dpe/df]\) but this convergence was slow. The GNN controller that replaces the integral controller should make the plant converge to the same steady-state vector, while limiting the duration and magnitude of the transients. Such an operation cannot be performed instantaneously. Besides, the value of the desired control action is not known beforehand.
Table 1

<table>
<thead>
<tr>
<th>Time</th>
<th>GNN controller</th>
<th>integral controller</th>
<th>Without controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.59</td>
<td>0.2666</td>
<td>0.3801</td>
<td>0.3736</td>
</tr>
<tr>
<td>1.17</td>
<td>0.0717</td>
<td>0.1876</td>
<td>0.3993</td>
</tr>
<tr>
<td>1.76</td>
<td>0.0617</td>
<td>0.1138</td>
<td>0.2318</td>
</tr>
<tr>
<td>2.34</td>
<td>0.0318</td>
<td>0.1029</td>
<td>0.2233</td>
</tr>
<tr>
<td>2.93</td>
<td>0.0005</td>
<td>0.0826</td>
<td>0.2550</td>
</tr>
<tr>
<td>3.54</td>
<td>0.0031</td>
<td>0.0626</td>
<td>0.2376</td>
</tr>
<tr>
<td>4.0996</td>
<td>0.0103</td>
<td>0.0393</td>
<td>0.3356</td>
</tr>
<tr>
<td>4.68</td>
<td>0.0013</td>
<td>0.0692</td>
<td>0.3356</td>
</tr>
<tr>
<td>5.27</td>
<td>0.0007</td>
<td>0.0397</td>
<td>0.2552</td>
</tr>
<tr>
<td>5.8690</td>
<td>0.0006</td>
<td>0.0231</td>
<td>0.2254</td>
</tr>
<tr>
<td>6.44</td>
<td>0.0000</td>
<td>0.0190</td>
<td>0.2553</td>
</tr>
<tr>
<td>7.02</td>
<td>0.0010</td>
<td>0.0149</td>
<td>0.2553</td>
</tr>
<tr>
<td>7.61</td>
<td>0.0000</td>
<td>0.0117</td>
<td>0.2553</td>
</tr>
<tr>
<td>8.19</td>
<td>0.0000</td>
<td>0.0092</td>
<td>0.2553</td>
</tr>
<tr>
<td>8.76</td>
<td>0.0000</td>
<td>0.0072</td>
<td>0.2553</td>
</tr>
</tbody>
</table>

Hence the simple, static, backpropagation algorithm is not directly applicable. It is necessary to modify according to the problem in hand and then refer back to backpropagation through time algorithm.

The dynamic GNN controller and plant structure is shown in Fig. 8. It is quite clear from the figure that the frequency can be sensed at every time instant. The GNN present controller output $u$ can be determined from the past value of $u$ and the frequency variation and the load perturbation. The load perturbation of large systems is not directly measurable. It must therefore be estimated by linear estimator or by non-linear neural network estimator. If the non-linearities in the system justify, such an estimator takes as inputs a series of $k$-samples of frequency fluctuations at the output of generator (d$f(t - 1)$, d$f(t - 2)$, ..., d$f(t - m)$) and estimates the instantaneous value of load perturbation d$p$ based on this input vector. The estimate d$p$ is then used to drive the plant controller. This can also be implemented with the help of GNN estimator. We assume that the electric perturbation is a step function of amplitude. When the step load perturbation hits the system the plant state changes which is necessary for control to be achieved by GNN controller.

The one way to implement GNN controller is to build a neural network emulator of the plant and backpropagating error gradients through it is nothing other than approximating the true Jacobian matrix of the plant using neural network training. Whenever the equations of the plant are known beforehand they can be used to compute analytically or numerically, the elements of the Jacobian matrix. The error gradient at the input of the plant is then obtained by multiplying the output error gradient by the Jacobian matrix. This approach avoids the introduction and training of neural network emulator which brings a substantial saving in development time. This approach is used by Beauchays [3].

The GNN controller uses the frequency variation samples for predicting the load disturbance and that load disturbance is used by GNN controller to control the plant dynamics.

Computer simulations have been conducted to illustrate the behaviour of single area system and two area system to step load perturbation and the performance of GNN controller is compared with the integral controller and with ANN controller. The results are tabulated in Table 1 and also shown in Fig. 9.

3.4. GNN Controller of a two area system

The GNN control scheme for a two area system is basically the same as for one area system. The level variable vector of two area system was $\{df_{1}, df_{2}, dp_{1}, dp_{2}\}$. After a step load perturbation has occurred in one area or simultaneously in both areas the level variable vector deviates from its steady-state value and the controller will control these variables, but the controller output is not instantaneous. The controller starts functioning after some delay time, so that the GNN estimator can estimate the load perturbation and that signal is available for the controller. The controllers of both areas start controlling when there is any variation in the frequency from its normal value. The performance of GNN controller of two area system is simulated on computer and the results are compared without controller and with integral controller as shown in Fig. 10. The results are also tabulated in Table 2. The performances of GNN controller is also tested under different values of regulation parameter $R$. It is seen that when the value of $R$ increases the frequency oscillations also increase, but die out after certain time. Beauchays [3] in his work controls the same plant dynamics using conventional ANN consisting of 20 hidden neurons and one output neuron. In the present work, the same plant dynamics can be controlled by one generalised neuron.

4. Conclusion

Generalised neural networks have been successfully applied to control the turbine reference power of a computer-simulated generator unit. The same principle has been applied to a simulated two area system. The GNN controllers have been adapted using backpropagation through time. In this chapter, the frequency variations in both areas of the two area system were put into the GNN controller. The GNN controller is found very suitable for controlling the plant dynamics in relatively less time. Each neural network controller receives only local information about the system (frequency in that specific area). Such an architecture decentralises the control of the overall system and reduces the amount of information to be exchanged between different modes of the power grid. Performance with local control only could be compared with that obtained with global control.
The successful application of this generalised neural network for the load frequency control of power system motivates to use this technique for estimation of load disturbance on the basis of frequency deviation (df) and voltage variations (dv) at the different buses. Also the generalised neural networks can be applied to a wide variety of problems in power system operation and control:

1. Maintenance of software and hardware in the control rooms of substations;

2. Power systems planning and designing;

3. Development of intelligent power systems controllers using GNN.

Knowledge-based systems can be used together with neural networks to handle the complex situations. One example which can be cited is the location of capacitors using artificial intelligence (AI) methods. Once the capacitors have been located, neural networks can be used for selecting the steps to put them into service. Further research,
therefore, must address the integration of AI based systems with neural networks.

Another problem is to co-ordinate HVDC controllers for multi-terminal or point to point systems by switching over among controllers. This problem can be resolved through a dual approach:

1. performance of tasks while the neural network learns;
2. learning through experience to improve system performance (neural networks).
Table 2
Performance of GNN controller for two area system

<table>
<thead>
<tr>
<th>Time (s)</th>
<th>Without controller</th>
<th>Integral controller</th>
<th>GNN controller</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>d1</td>
<td>d2</td>
<td>d1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.5000</td>
<td>-0.0173</td>
<td>-0.1332</td>
<td>-0.0126</td>
</tr>
<tr>
<td>1.0000</td>
<td>-0.0683</td>
<td>-0.1492</td>
<td>-0.0614</td>
</tr>
<tr>
<td>1.5000</td>
<td>0.1097</td>
<td>0.1207</td>
<td>-0.0694</td>
</tr>
<tr>
<td>2.0000</td>
<td>-0.1248</td>
<td>0.1091</td>
<td>-0.0307</td>
</tr>
<tr>
<td>2.5000</td>
<td>-0.1237</td>
<td>-0.1113</td>
<td>-0.0063</td>
</tr>
<tr>
<td>3.0000</td>
<td>-0.1191</td>
<td>-0.1161</td>
<td>0.0021</td>
</tr>
<tr>
<td>3.5000</td>
<td>-0.1168</td>
<td>-0.1184</td>
<td>-0.0047</td>
</tr>
<tr>
<td>4.0000</td>
<td>-0.1168</td>
<td>-0.1185</td>
<td>-0.0056</td>
</tr>
<tr>
<td>4.5000</td>
<td>-0.1173</td>
<td>-0.1180</td>
<td>-0.0027</td>
</tr>
<tr>
<td>5.0000</td>
<td>-0.1177</td>
<td>-0.1176</td>
<td>-0.0014</td>
</tr>
<tr>
<td>5.5000</td>
<td>-0.1178</td>
<td>-0.1175</td>
<td>-0.0019</td>
</tr>
<tr>
<td>6.0000</td>
<td>-0.1177</td>
<td>-0.1176</td>
<td>0.0022</td>
</tr>
<tr>
<td>6.5000</td>
<td>-0.1176</td>
<td>-0.1176</td>
<td>-0.0017</td>
</tr>
<tr>
<td>7.0000</td>
<td>-0.1176</td>
<td>-0.1177</td>
<td>-0.0012</td>
</tr>
<tr>
<td>7.5000</td>
<td>-0.1176</td>
<td>-0.1177</td>
<td>-0.0010</td>
</tr>
<tr>
<td>8.0000</td>
<td>-0.1176</td>
<td>-0.1176</td>
<td>-0.0009</td>
</tr>
<tr>
<td>8.5000</td>
<td>-0.1176</td>
<td>-0.1176</td>
<td>-0.0008</td>
</tr>
<tr>
<td>9.0000</td>
<td>-0.1176</td>
<td>-0.1176</td>
<td>-0.0006</td>
</tr>
<tr>
<td>9.5000</td>
<td>-0.1176</td>
<td>-0.1176</td>
<td>0.0005</td>
</tr>
</tbody>
</table>

References