Fuzzy-PLC PID Simulink implemented AVR system to enhance the transient response of synchronous generator

ABDULLAH J. H. AL GIZI¹, M.W. MUSTAFA¹
¹Faculty of Electrical Engineering
Universiti Teknologi Malaysia
81310 JOHOR, MALAYSIA

Corresponding author: E-mail addresses: abdullh969@ yahoo.com

Abstract: - A novel design a fuzzy proportional integral derivative (PID) controller is using Matlab and programmable logic controllers (PLCs) for a set point voltage control problem in the automatic voltage regulator (AVR) system. The controller objective is to maintain the terminal voltage all the time under any loads and operational conditions by attaining to the desired range via the regulation of the generator exciter voltage. The main voltage control system uses PLCs to implement the AVR action. The proposed fuzzy controller combines the genetic algorithm (GA), radial-basis function network (RBF-NN) identification and fuzzy logic control to determine the optimal PID controller parameters in AVR system. The RBF tuning for various operating conditions is further employed to develop the rule base of the Sugeno fuzzy system. The fuzzy PID controller FPID is further designed to transfer in PLCs (Step 75.5) for implementing the AVR system to maintain the terminal voltage all the time under any loads and operational conditions for synchronous generator constant.

Key-Words: - Fuzzy logic, FPID, PLC, Step 75.5, Transient response

1 Introduction
Automation control systems are widespread in science and technology. The typical hardware device used in engineering control is Programmable Logic Controllers (PLC) that controls numerous industrial systems. Call of Simulink subsystems from MATLAB to STEP 75.5 for implementing PLC use a number of control loops responsible for upholding the actions excellence of the process are considerably increased with ever-growing difficulties of modern process plants. As a result, the modelling and simulation of the control systems became multifaceted. The major challenges in compound system are the extreme nonlinearities and the interaction between the control loops that make modelling difficult. PLC is exploited to control plants or industrial equipment’s such as water and waste control, energy, oil and gas refining, etc. to cite a few [1]. Generally, a fuzzy PID controller is developed using PLC for tackling the problem of a set point pressure control in the main pressure collection system. An intelligent hierarchical coordinated control strategy is successfully applied by Hongbo et al. to a 300 MW boiler-turbine unit in China [2]. The theory introduced by Zadeh deals with the doubt and fuzziness related information concerning several parameters [3].The main objective of the AVR system is to control the terminal voltage by adjusting the generator exciter voltage. It must keep track of the generator terminal voltage all the time under any load condition by maintaining the voltage within pre-determined limits [4]. Despite much efforts in developing advanced control schemes, the control of classical integrated PLC-Fuzzy PID Simulink implemented AVR system is far from being understood [5]. The PID possessing differential, proportional and integral coefficients optimally controls the AVR system. Computational techniques such as GA and fuzzy logic are used for analytic solution[6]. A tuning fuzzy logic approach for determining the optimal PID controller parameters in AVR system is developed to obtain on-line PID parameters under various operating conditions [7].Minglin designed a PID-like fuzzy controller with FPGAv [7]. The feed forward fuzzy PID controller is used to improve the performance of high pressure common rail system [8]. Sinthipsomboon et al. developed a hybrid fuzzy self-tuning PID controller to enhance the performance by suitably adjusting the system parameters [9]. An improved Fuzzy PID controller is used to control Brushless DC motor speed [10]. The design and detailed stability analysis of Takagi-Sugeno-Kang (TSK) type full-scale fuzzy PID controller is demonstrated [10]. Parameter self-setting fuzzy PID algorithm for controlling the fluctuations and improving the drying temperature is reported [11]. A self-tuning PID tracking controller.
based on RBF neural network with fuzzy current limiter is developed to maneuver the motor and save energy [12]. An approach is used for controlling PMSM servo system using fuzzy radius basis function (f-RBF) neural network that acquire the advantages of strong adaptive ability and nonlinear approximation capability [13]. Zhang et al. designed a new algorithm of vehicle stability adaptive PID control with single neuron network suitable for implementing the real time manipulation [14, 15]. Kun et al. employed radical basis function (RBF) to develop an optimal PID controller called direct-drive permanent magnet linear synchronous motor (PMSM) [16]. They can easily be understood, maintained by field engineer and can be combined with Sugeno fuzzy logic model rule to obtain RBF-NN tuning via genetic algorithm for designing FPID controller. Furthermore, call of Simulink subsystems from MATLAB to STEP 75.5 can be integrated to PLC [17]. PLC control system is specially designed for industrial environment application with excellent stability and reliability. The attractive features of PLC such as simple, flexible, easy system configuration with low cost, low maintenance and running cost make them suitable for implementation. PLC being a specialized computer which interfaces a set of inputs to sensors and a set of outputs to actuators can control the plant by performing various functions such as logic, sequencing, timing, counting and arithmetic. PLC control system is a versatile system consisting of several PLCs and computers coupled together for operation. We report a novel design method by integrating the Sugeno fuzzy system rule base and the AVR system fuzzy PID controller FPID with the STEP 75.5 by combining PLCs as a hardware control unit for maintain the terminal voltage per-establish limits under any loads and operational conditions.

2 Problem Formulation

Hardware control unit for maintain the terminal voltage all the time under any loads and operational conditions for synchronous generator constant by design FPID with the STEP 75.5.

2.1 Radial Basis Function Networks

The outer loop of AVR is a self-tuning PID voltage controller based on the radial basis function neural network that has an ability to adapt with uncertain load and system conditions. Moody et al. proposed a feed-forward two-layered RBF neural network with single hidden layer to mimic the systematic arrangement of restrictive readjustment in the human mind [12]. Furthermore, the input samples for RBF neural network do not require a special distribution and RBF possess an on-line learning with rapid converges. Consequently, the control field for implementing the real time manipulation concentrates on the neural network. The RBF is exploited to achieve the best parameters of the controller to maintain zero system error [12]. The schematic of radial-basis function neural network is shown in Fig.1. The updating algorithm for the adaptive PID based RBF can be formulated as,

\[ \Delta k_p = \mu . e(k).e_p(k) \sum_{j=1}^{w} w_j h_j \frac{c_{ij} - u(k)}{\sigma_j} \]  

(1)

\[ \Delta k_i = \mu . e(k).e_i(k) \sum_{j=1}^{w} w_j h_j \frac{c_{ij} - u(k)}{\sigma_j} \]  

(2)

\[ \Delta k_d = \mu . e(k).e_d(k) \sum_{j=1}^{w} w_j h_j \frac{c_{ij} - u(k)}{\sigma_j} \]  

(3)

The PID parameters such as integral gain (Ki), the proportional gain (Kp) and the derivative gain (Kd) are automatically readjusted by RBF on-line learning algorithm to maintain the system error \( e(k) = 0 \). Two commands offered by Matlab namely Newrb and newrbe are used to design the RBF neural network in which Newrb adds neurons step by step until the goal is hit with long training time with minimal error and newrbe designs a network very quickly with zero error [18, 19]. In the training process, the achieved steps are: (i) neurons number in the hidden layer, (ii) the coordinates of the center of RBF function (iii) and the radius (spread) of each RBF functions in each dimension.

![Fig.1 Schematics of RBF neural network structure.](image-url)
maintain constant terminal voltage at different levels. The transfer function of the AVR components is summarized in Table 1.

![Block diagram of AVR system along with PID controller](image)

Fig.2 Block diagram of AVR system along with PID controller.

<table>
<thead>
<tr>
<th>Component</th>
<th>Transfer function</th>
<th>Parameter limits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amplifier</td>
<td>$TF_{amplifier} = K_s / (1 + \tau_s s)$</td>
<td>$10 &lt; K_s &lt; 40$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.02 s &lt; \tau_s &lt; 1 s$</td>
</tr>
<tr>
<td>Exciter</td>
<td>$TF_{exciter} = K_i / (1 + \tau_i s)$</td>
<td>$1 &lt; K_i &lt; 10$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.4 s &lt; \tau_i &lt; 1 s$</td>
</tr>
<tr>
<td>Generator</td>
<td>$TF_{generator} = K_p / (1 + \tau_p s)$</td>
<td>Kg depend on the load</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.01 s &lt; \tau_p &lt; 2 s$</td>
</tr>
<tr>
<td>Sensor</td>
<td>$TF_{sensor} = K_r / (1 + \tau_r s)$</td>
<td>$0.001 s &lt; \tau_r &lt; 0.06 s$</td>
</tr>
</tbody>
</table>

The $MinF(K_p, K_i, K_r)$ combines transient response counting rise time overshoot, settling time and steady-state error. The satisfaction of the designer needs can be achieved by choosing suitable value of the weighting factor $\beta$. Therefore, the optimization problem boils down to the following constraints,

$$K_p^{min} \leq K_p \leq K_p^{max}, K_i^{min} \leq K_i \leq K_i^{max}, K_r^{min} \leq K_r \leq K_r^{max} \quad (7)$$

Following Devaraj et al. [20], RGA is applied to optimize the values controller parameters and the proposed GA is introduced.

### 2.3 Proposed GA

GA is recognized as an effective and efficient technique to solve the optimization problems. In comparison to the optimization techniques, such as random search and simulated annealing, GA performance is superior that avoids local minima considered as a key issue in nonlinear systems [6, 7].

#### 2.3.1 Genetic Algorithm Operators

The genetic algorithms are based on the natural selection mechanism that allows survival of the fittest and generate estimated solutions by exchanging information’s to attain the optimum solution. After generating the initial population, the GA discovers new individuals by producing offsprings using the reproduction, crossover and mutation operators, which replace the old generation members and form the new generation. Once several generations are produced, the algorithm finds the best chromosome that represents the optimum or near optimum solution. The major GA operators such as cross-over, reproduction and mutation are exploited. The convergence speed is controlled by applying various probabilities on these operators. The design of the crossover and mutation operators are carefully managed due to their immense impact on the performance of genetic algorithm [6, 7]. The details of the genetic operators used in the proposed GA are illustrated in Table 1.

#### 2.3.1.1 Reproduction

In the process of reproduction, individuals are selected depending on their fitness function, the higher the fitness is, more chance for an individual to be selected for the next generation. Three main selection methods such as ranking method, fitness balanced selection and tournament selection are utilized [20]. In this work, we employ the tournament selection method, where ‘n’ individuals are randomly selected from the population and the
best vale is chosen for additional genetic processing. This process is repeatedly performed until the mating pool is filled.

2.3.1.2 Crossover

The property of global search in GA is mostly determined by the crossover operator, which combines two-parent chromosomes to produce a new one. The range of the selected probability is typically between 0.6 – 1.0. One of the interesting features of the crossover operators is the relation between the generated chromosome and the location of both the parents. The generated new chromosome remains close to the parents in case both the parents are close to each other. Conversely, the search is more likely to be random [20].

2.3.1.3 Mutation

New chromosome is inserted into the population for the mutation process. Mutation randomly makes insignificant change in the chromosome information. However, for inconsistent mutation, the variable takes a consistent random number between the lower and upper limits. In this study ‘uniform mutation’ operator is used.

2.4 GA Implementation for Optimizing PID Parameters of AVR

The optimum PID controller parameters are obtained via GA tuning of PID. Two major themes such as symbol of the choice variables (variable representation) and arrangement of the fitness function are used in this process.

2.4.1 Variable Representation

The solutions of all candidates are generated in the genetic population. The solution elements of PID controller-tuning problem include parameters \(K_i\), \(K_p\) and \(K_d\). The direct representation of the solution variables reduces the computer space for storing the population. The values of these parameter obtained from direct tuning of GA into the RBF program for the optimum tuning of the PID controller parameter are substantial for the thematic factory operation of AVR system.

2.4.2 Fitness Function

The solution for the performance of every candidate in the population is evaluated based on its fitness which is defined as a non-negative value to be maximized. Fitness is associated in a straight line with the value of objective function. The parameter set of the individual evaluation can be determined using equation (6) for the performance criteria. The value of individual fitness is calculated by the outcome of the presentation criteria via mutual computation. The fitness function is the presentation of mutuality criterion \(F(K_p,K_i,K_d)\) given in equation (6). Thus, the minimization of performance criteria in eq. (6) can be transformed to the maximization of the fitness function as,

\[
Fitness = \frac{k}{F(K_p,K_i,K_d) * ITAE}
\]

Where \(k\) is a constant, ITAE is a time integral multiplied by the absolute error value. This is used to amplify the value of 1/F, which is generally small, so that, the chromosome fitness values occur in a wider range.

3 Sugeno Fuzzy Model

Recently, Devaraj et al. used fuzzy set theory, in which a variable is a member of one or more sets, with a membership specified degree [20]. The fuzzy rule is expressed as,

\[
If \ x \ is \ A \ and \ y \ is \ B \ then \ z = f(x,y)
\]

where A and B are fuzzy sets in the antecedent, \(x\) and \(y\) are input variables and \(f(x,y)\) is a crisp function in the consequent. Each variable fuzzy set are represented by suitable membership functions. The core of the fuzzy logic system is formed by a set of such rules. For an exact input signal condition, the fuzzy system defines the rules to be fired and then calculates the efficient output in two steps. Firstly, the minimum of the membership functions input\(w_i\) is obtained for each rule, where this value is the firing value for a particular rule. Secondly, the overall output is calculated by a weighted average of individual rule outputs given by,

\[
z = \sum_{i=1}^{N} \frac{\mu_i z_i}{\sum_{i=1}^{N} \mu_i}
\]

The PID controller parameters under various operating conditions are determined by the Sugeno fuzzy system.

4 Problem Solution

Design novel FPID by combining the RBF-NN, GA and Sugeno fuzzy logic to determine the optimal parameters of PID controller
integrating the with the STEP 75.5 PLCs as a hardware control unit.

4.1 Mythology to design fuzzy PLC PID controller

4.1.1 Development of a Sugeno Fuzzy Model to Design PID controller

The optimum PID parameters for real-time operation are obtained by developing Sugeno fuzzy logic model with \( K_e \) and \( \tau_e \) as inputs and \( K_p \), \( K_i \) and \( K_d \) as outputs. As much as eight fuzzy sets such as ‘very low (VL)’, ‘low (L)’, ‘medium low (ML)’, ‘medium (M)’, ‘medium high (MH)’, ‘high low (HL)’, ‘high medium (HM)’, and ‘high (H)’ are defined for the variable \( K_e \). Likewise, six fuzzy sets defined for the variable \( \tau_e \) ‘very low (VL)’, ‘low (L)’, ‘medium low (ML)’, ‘medium high (MH)’, ‘high (H)’ and ‘very high (VH)’.They are linked with overlapping triangular membership functions. To formulate the table for fuzzy rule, the values of \( K_e \) are varied from 2.0 to 9.0 in steps of 1.0 and \( \tau_e \) are varied from 0.5 to 1 in steps of 0.1. For each combination of \( K_e \) and \( \tau_e \), the proposed RBF tuning via GA is applied to obtain the optimal values of \( K_p \), \( K_i \) and \( K_d \) in each times. The fuzzy rules formulated for \( K_p \), \( K_i \) and \( K_d \) are listed in Table 2 (a), (b) and (c), respectively. During real-time operation, corresponding to the current operating conditions, the values of \( K_e \) and \( \tau_e \) are determined. For these values of \( K_e \) and \( \tau_e \), the optimal value of \( K_p \), \( K_d \) and \( K_i \) can be computed using the fuzzy rule table and the FIS editor Sugeno inference system explained in section 4.2. Depending on the initialization (FIS editor), the fuzzy logic controller inputs are \( K_e \), \( \tau_e \) and outputs are (\( K_p \), \( K_d \) and \( K_i \)). The system with three fuzzy logic controllers (\( K_p \), \( K_d \) and \( K_i \)) with rule viewer are set in which each controller has two inputs (\( K_e \), \( \tau_e \)) and each input has associated fuzzy set. The output has 144 fuzzy set rules for \( K_p \), \( K_d \) and \( K_i \) and 48 rules for each one parameter as depicted in Fig.3.

Table 2(a) the fuzzy rule table formulated for \( K_p \) using the above approach

<table>
<thead>
<tr>
<th>( \tau_e )</th>
<th>Very low</th>
<th>Low</th>
<th>Medium Low</th>
<th>Medium high</th>
<th>High</th>
<th>Very high</th>
</tr>
</thead>
<tbody>
<tr>
<td>( K_e )</td>
<td>0.5</td>
<td>0.6</td>
<td>0.7</td>
<td>0.8</td>
<td>0.9</td>
<td>1</td>
</tr>
</tbody>
</table>

(a) For proportional gain \( K_p \)

<table>
<thead>
<tr>
<th>( K_p )</th>
<th>Very Low(2)</th>
<th>Low (3)</th>
<th>Medium low (4)</th>
<th>Medium High (6)</th>
<th>High (7)</th>
<th>Medium medium (8)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.2944</td>
<td>0.4153</td>
<td>0.2859</td>
<td>0.1039</td>
<td>0.1164</td>
<td>0.1780</td>
</tr>
<tr>
<td></td>
<td>0.6124</td>
<td>0.3768</td>
<td>0.3304</td>
<td>0.2158</td>
<td>0.1123</td>
<td>0.2422</td>
</tr>
<tr>
<td></td>
<td>0.5980</td>
<td>0.4796</td>
<td>0.3377</td>
<td>0.1085</td>
<td>0.2108</td>
<td>0.2486</td>
</tr>
<tr>
<td></td>
<td>0.4622</td>
<td>0.5115</td>
<td>0.3617</td>
<td>0.3141</td>
<td>0.1747</td>
<td>0.2615</td>
</tr>
<tr>
<td></td>
<td>0.4728</td>
<td>0.4887</td>
<td>0.3662</td>
<td>0.3099</td>
<td>0.2715</td>
<td>0.2570</td>
</tr>
<tr>
<td></td>
<td>0.6079</td>
<td>0.3817</td>
<td>0.3835</td>
<td>0.3477</td>
<td>0.2037</td>
<td></td>
</tr>
</tbody>
</table>

Table 2(b) the fuzzy rule table formulated for \( K_i \) using the above approach

<table>
<thead>
<tr>
<th>( \tau_e )</th>
<th>Very low</th>
<th>Low</th>
<th>Medium Low</th>
<th>Medium high</th>
<th>High</th>
<th>Very high</th>
</tr>
</thead>
<tbody>
<tr>
<td>( K_i )</td>
<td>0.5</td>
<td>0.6</td>
<td>0.7</td>
<td>0.8</td>
<td>0.9</td>
<td>1</td>
</tr>
</tbody>
</table>

(b) For integral gain \( K_i \)

<table>
<thead>
<tr>
<th>( K_i )</th>
<th>Very Low(2)</th>
<th>Low (3)</th>
<th>Medium low (4)</th>
<th>Medium High (6)</th>
<th>High (7)</th>
<th>Medium medium (8)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.4201</td>
<td>0.2960</td>
<td>0.2424</td>
<td>0.2193</td>
<td>0.1902</td>
<td>0.1749</td>
</tr>
<tr>
<td></td>
<td>0.4280</td>
<td>0.3060</td>
<td>0.2427</td>
<td>0.2112</td>
<td>0.1947</td>
<td>0.1860</td>
</tr>
<tr>
<td></td>
<td>0.4276</td>
<td>0.3218</td>
<td>0.2638</td>
<td>0.2370</td>
<td>0.2055</td>
<td>0.1840</td>
</tr>
<tr>
<td></td>
<td>0.4718</td>
<td>0.3372</td>
<td>0.2671</td>
<td>0.2362</td>
<td>0.2114</td>
<td>0.1937</td>
</tr>
<tr>
<td></td>
<td>0.5110</td>
<td>0.3594</td>
<td>0.2798</td>
<td>0.2397</td>
<td>0.2190</td>
<td>0.2035</td>
</tr>
<tr>
<td></td>
<td>0.5277</td>
<td>0.3739</td>
<td>0.2946</td>
<td>0.2415</td>
<td>0.2271</td>
<td></td>
</tr>
</tbody>
</table>

Table 2 (c) The fuzzy rule table formulated for \( K_d \) using the above approach

<table>
<thead>
<tr>
<th>( \tau_e )</th>
<th>Very low</th>
<th>Low</th>
<th>Medium Low</th>
<th>Medium high</th>
<th>High</th>
<th>Very high</th>
</tr>
</thead>
<tbody>
<tr>
<td>( K_d )</td>
<td>0.5</td>
<td>0.6</td>
<td>0.7</td>
<td>0.8</td>
<td>0.9</td>
<td>1</td>
</tr>
</tbody>
</table>

(c) For derivative gain \( K_d \)

<table>
<thead>
<tr>
<th>( K_d )</th>
<th>Very Low(2)</th>
<th>Low (3)</th>
<th>Medium low (4)</th>
<th>Medium High (6)</th>
<th>High (7)</th>
<th>Medium medium (8)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.0241</td>
<td>0.1733</td>
<td>0.0697</td>
<td>0.0144</td>
<td>0.0877</td>
<td>0.0710</td>
</tr>
<tr>
<td></td>
<td>0.0378</td>
<td>0.1436</td>
<td>0.0199</td>
<td>0.0106</td>
<td>0.0160</td>
<td>0.0179</td>
</tr>
<tr>
<td></td>
<td>0.0540</td>
<td>0.1489</td>
<td>0.0207</td>
<td>0.0391</td>
<td>0.0388</td>
<td>0.0658</td>
</tr>
<tr>
<td></td>
<td>0.2007</td>
<td>0.0285</td>
<td>0.1572</td>
<td>0.0400</td>
<td>0.0391</td>
<td>0.0661</td>
</tr>
<tr>
<td></td>
<td>0.1916</td>
<td>0.0284</td>
<td>0.0391</td>
<td>0.0221</td>
<td>0.0175</td>
<td>0.0191</td>
</tr>
<tr>
<td></td>
<td>0.2102</td>
<td>0.2074</td>
<td>0.1035</td>
<td>0.0904</td>
<td>0.1015</td>
<td>0.0168</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.0146</td>
<td>0.0632</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.0136</td>
<td>0.0398</td>
</tr>
</tbody>
</table>

Fig.3 Surface and rule viewer for FPID controller.
2.1 Experimental

Fig. 4 depicts the modified model suitable for studying the transient stability response. The test model including a synchronous generator (400 V 3-phase 30 KVA, PF 0.8, 45.6 A and 1500 RPM) coupled with par mover internal composition gas engine (Kia motors 2701), control circuit (PLC type SIMATIC IPC427C attached with gearbox, convertor, speed sensor, voltage sensor, AVR 400V 10A), measurement devices (voltmeter, frequency meter and digital oscilloscope), supplementer devices, transient device, load resistances, step down transformers 220-6 V, CB 3-phase 60 A and two DC power supplies (0 - 5, 0 - 30) (0 - 24) are utilized.

Fig. 4 Experimental set up for testing the FPID controller in AVR systems.

2.1.1 System Description

The design for FPID data transfer from MATLAB to PLC is carried out in the following way:

2.1.2 C/C++ Generation with RTWEC

RTWEC is an add-on of Math works which generates C/C++ code from Simulink subsystems and opens the Simulink model. In the menu by clicking on "Tools – Real-Time Workshop- Options..." a window is displayed with a navigation bar. The entry "Real-Time Workshop" is selected by default.

The following parameters are required to enter: Navigation item "Real-Time Workshop" – System target file: ert.tlc – Language: C++ – Click the button "Set objectives" Shift "Traceability" and "Execution efficiency" with the "->" button to the right field. Navigation item "Code Placement" – File packaging format: Compact. With the right mouse-button click on the subsystem "PID_FPID_disc", Select "Real-Time Workshop Build Subsystem..." in the context menu, in the window "Build code for Subsystem" click on the "Build" button. Once the C/C++ code is created, the window automatically closes "Build Code for Subsystem". The generated code is located in the directory of the Simulink model"..\PID_FPID_disc_rtw" as shown in Fig. 5.

Fig.5 Typical code for designing PID_FPID controlled systems

2.1.3 WinAC Simulink to ODK (S2O) Wizard

WinAC S2O Wizard automatically generates all required blocks and files for the integration of the Simulink subsystem into a STEP 7 project. An SCL source and a DLL or RTDLL file is created from the generated C/C++ code of RTWEC. The integration is performed with STEP 7 V5.x programs through WinAC S2O Wizard. Finally, the loading of STEP 7 program with DLL or RTLL are administered to a PC system using WinAC RTX. In addition, the WinAC ODK Library is integrated in the SIMATIC Manager that requires blocks SFB65001 (CREA_COM) and SFB65002 (EXEC_COM) for executing DLL/RTDLL. Fig.6 The FB
PID_FPID_disc” into the "Network 2: PID Controller.

2.1.4 Configuration with SIMATIC Manager

STEP 7 V5.5

The folder "… \ FPID_STEP7_V5x_Project" contains the project for STEP7 V5.5 with the following contents:

i. WinAC RTX is the only the hardware configuration of PC station with WinAC RTX serves as a template and the program is empty.

ii. PID_FPID provides a prepared program for integration through the WinAC S2O Wizard. The following blocks are simultaneously created:

- OB35 (CYC_INT5) as cyclic OB with 100 ms cycle. The simulated controlled system and the PID controller (PID_FPID_disc) are called in this block.
- DB35 (Data) as a global data block that contains all required variables such as Setpoint [Real] and PID_output [Real].
- Process_value [Real], crea_status [Word] (provides the status via CREA_COM), exec_status [Word] (provides the status via EXEC_COM) and initialize [Bool].
- FB100 (PROC_C) with Instanz-DB100 Simulated PT3 process (parameterized like the process in MATLAB/Simulink).
- SFB65001 (CREA_COM) for initialization of the DLL/RTDLL file.

iii. PID_FPID_dll_final contains a complete programming with integration of the PID controller from MATLAB/Simulink through the WinAC S2O Wizard. A DLL call is used for this program.

iv. PID_FPID_rtdll_final include a complete programming with integration of the PID controller from MATLAB/Simulink through the WinAC S2O Wizard. An RTDLL call is used in this program. The steps required to complete the program "PID_FPID" with the "PID_FPID_disc" block from the WinAC S2O Wizard are depicted in Fig.6.

2.2 FPID-PLC Controller

The PLC (FPID-PLC) controller acquires the control signal (0 - 10 V) from voltage sensor that connects between any two lines (R and S) of generator terminal voltage.

The control signal depends on the voltage difference between line to line (R-S or S-T or R-T) generator terminal to FPID-PLC controller. The field excitation resistance (excitation current) is varied precisely via a gearbox (gearbox to make the motion of change variable resistance smooth and accurate). Conversely, the speed data control signal (4 - 20 mA) originates from speed sensor that fixes the shaft generator by coupling it to internal composition KIA 2701 diesel engine of Industrial controller KS 92 to maintain the speed constant at 1500 rpm though the gearbox.

The integration between the generator rotor speed and excitation current for obtaining the terminal voltage at constant frequency. The hardware installation is carried out using SIMATIC IPC427C (MICROBOX PC) and S7 program is loaded with the specified configuration such as PG/PC interface: Ethernet (192.168.2.200) and IPC427C interface: PROFINET CP1616 (192.168.2.10). The three types of AVR used by us are shown in Fig.7. For comparison of our results, AVR 1 is used with FPID-PLC controller but AVR2 and AVR3 are employed without FPID-PLC controller.
Fig. 7 Three types of AVR used

It is a vital matter for the stable electric power service to develop the AVR of the synchronous generator with a high potency and a quick response. To analyse the performance of the AVR system under severe fault, 3-phase to ground fault is applied at the generator terminal and the system response is observed. The system response for above contingency with FPID is achieved. The fuzzy PID controller as described above is evaluated by means of practical experiments with main voltage system. The collecting excitation current in the experiment is controlled by using the fuzzy PID control algorithm. The terminal voltage is independently adjusted and the obtained results are presented in Figs. 8, 9, & 10. Whereas, these Figs. showen the different between used the FPID controller with AVR and used the AVR without the novel controller. The AVR1 system adjusted by FPID-PLC controller keeping the terminal voltage constant is shown in Fig. 8(a). Fig. 8(b) represents the system response for above contingency with FPID-PLC. It is clearly seen that the FPID-PLC is able to suppress the oscillation in the terminal voltage owing superior damping characteristics as compared to AVR2 and AVR3 without controller and canceling the overshoot and minimize the swing. The ability of FPID in suppressing the oscillation of the terminal voltage and good damping characteristic compression is clearly noticeable [20-22]. Fig. 9 & 10 clearly demonstrates how the voltages swing with AVR2 and AVR3 under severe disturbance. The poor ability AVR2 and AVR3 in suppressing the oscillation of the terminal voltage and damping characteristic compression with AVR1.

Fig. 8 (a) & (b) Voltage control curve for AVR1 adjusted with FPID and Voltage control curve for severe disturbance.

Fig. 9(a)&(b) Voltage control curve for AVR2 adjusted without FPID and Voltage control curve for severe disturbance.

Fig. 10(a)&(b) Voltage control curve for AVR3 adjusted without FPID and Voltage control curve for severe disturbance.
The voltage for AVR2 and AVR3 swing between 600 - 150 V and 550 - 152 V, respectively, in comparison to AVR1 where it ranges between 200 - 410 V as shown in Fig.11 clearly demonstrates how the voltages swing with AVR2 and AVR3 under severe disturbance. It is clearly seen that the FPID-PLC is able to suppress the oscillation in the terminal voltage owing superior damping characteristics as compared to AVR2 and AVR3 without controller as shown in Fig.11.

Fig.11 Voltage response under severe disturbance of AVR1, AVR2 and AVR3.

3 Conclusion

We develop a novel combined approach of GA, Sugeno fuzzy logic and RBF-NN to achieve the optimal PID controller parameters in AVR system. This new fuzzy PID control approach with a PLC is proposed to improve the voltage control performance related to inherent interacting effect in the collecting control loops. Experimental results demonstrate the significant improvement in attaining the desired voltage range using the proposed fuzzy PID controller to provide good control performance at various operating conditions. The achieved accurate control the voltage response under severe disturbance of AVR1, AVR2 and AVR3 suggest that our method constitutes a basis for improving the transient response of power generation systems using automation control suitable for wide-spread applications.

References:

[17] Siemens SIMATIC WinAC S20 Wizard February 2012