Fine Tuning of Fuzzy Rule-Base System and Rule Set Reduction Using Statistical Analysis

Muhammad Babar Nazir
Shaoping Wang

Department of Mechatronic, School of Automation Sciences and Electrical Engineering, Beihang University, Beijing 100083, China

1 Introduction

Fuzzy rule-based systems (FRBSs) can describe complex systems in terms of linguistic descriptions [1] and do not require complex mathematical modeling. Thus, fuzzy logic control (FLC) is going to be a conventional control method since the control design strategy is simple and practical. The quantitative relationship between variables in linguistic terms can be described by fuzzy rule base [2]. However, the major obstacle in the application of fuzzy systems is derivation of such rule base, which truly can describe the behavior of the real system [3]. Numerous techniques—based on genetic algorithms (GAs) [3–8], based on multi objective evolutionary algorithm [9] evolutionary programming [10] and evolutionary algorithms [11], based on co-evolution of subpopulations [12], based on neural network [13]—have been developed to contribute in this regard.

The key concern in the design of FRBSs is how to achieve an optimal balance between interpretability and accuracy. Interpretability, capability to express the behavior of the real system in a comprehensible way, and accuracy, capability to faithfully represent the real system, are the key aspects having contradictory requirements. Hence, improvement in one of these aspects generally causes worsen of the other. Mamdai based FRBSs, linguistic fuzzy model (FM), mainly emphasize on the interpretability whereas, Takagi–Sugeno FRBSs, Precise FM, focus on the accuracy [3]. Many techniques have been developed for improvement in the FRBSs. The techniques for the accuracy improvement of linguistic FM focused on learning and tuning of rule base [3–8,14], tuning of membership functions [3,8,15–17], modifying the membership functions [18], and extended tuning using linguistic hedges [3,9]. The techniques for interpretability improvement focused on reducing the rule base [3,9,19], and reducing the number of input variables [20].

The optimal balance between interpretability and accuracy is the crucial issue need to be addressed. Techniques for fuzzy modeling as mentioned above cannot achieve the optimal balance. Therefore, the key objective of this research work is to achieve an optimal balance between interpretability and accuracy for linguistic FRBSs. The rule base is automatically derived using genetic algorithm from the numerical input and output data sets. The accuracy of FRBSs is enhanced by simultaneous tuning of scaling factors, MFs, and linguistic hedges using novel evolutionary algorithm [21] due to its computational efficiency and rapid convergence. Tuning of scaling factors ensures the mapping of variables into universe of discourse and tuning of MFs ensures the fine distribution and optimal overlapping of MFs to improve the accuracy. Linguistic hedges are used to preserve the interpretability requirement. A statistical based rule reduction technique is developed to further improve the interpretability. A modified reduced ordered rule base is developed emphasizing on granularity of partition.

The proposed technique is applied for nonlinear electrohydraulic servo system. LuGre dynamic friction model is incorporated in the system model to account for main nonlinear effects. Extensive simulation and experiment results authenticate the validity and practicability of proposed technique under parametric uncertainties and external disturbances.

The rest of paper is organized as follows: Sec. 2 shows the development of mathematical model for nonlinear electrohydraulic servo system (EHSS). Section 3 represents the development of fuzzy proportional-integral-derivative (PID) controller for EHSS. Section 4 gives the autogeneration of fuzzy rules from numerical input and output data sets. Section 5 gives the detail discussion of simultaneous tuning of scaling factors, MFs, and linguistic hedges using NEA. Section 6 develops the technique for rule reduction based on statistical approach. Section 7 discusses the extensive simulation results and practical application is given in Sec. 8. Based on the meticulous analysis of these results, conclusion is made in Sec. 9.

Keywords: learning fuzzy rule, fine tuning, novel evolutionary algorithm, rule reduction, interpretability, statistical analysis
2 Mathematical Modeling of Nonlinear EHSS

EHSS consists of hydraulic power supply, Moog servo valve, hydraulic motor, and encoder shown in Fig. 1.

The mathematical model of nonlinear EHSS is developed under the following assumptions [22,23]:

(a) The supply pressure is constant.
(b) Servo valve orifices are symmetrical.
(c) Servo valve dynamics are neglected.
(d) Motor external leakage is negligible.

The nonlinear load flow of servo valve neglecting leakage is given as

\[ Q_f = C_w x \frac{p_f - \text{sgn}(x) p_f}{\rho} \]  

(1)

where \( Q_f \) is the load flow, \( C_w \) is the coefficient of discharge, \( W \) is the area gradient, \( x \) is the displacement of the spool, \( p_f \) is the supply pressure, \( p_f \) is the load pressure, and \( \rho \) is the fluid mass density.

It is obvious that the load flow of servo valve is nonlinear with absolute value of load pressure.

The torque balance equation is given as

\[ D_m p_f = J_m \frac{d^2 \theta_m}{dt^2} + B_m \frac{d \theta_m}{dt} + G \theta_m + T_f \]  

(2)

where \( D_m \) is the volumetric displacement of motor, \( \theta_m \) is the angular position of motor, \( V_m \) is the total volume of motor chamber, \( E_m \) is the effective bulk modulus, \( C_m \) is the leakage coefficient of motor, \( J_m \) is the inertia of motor, \( B_m \) is the viscous damping coefficient of load, \( G \) is the equivalent torsional spring gradient of load, and \( T_f \) is the friction torque.

Usually, friction is the most important nonlinear factor that influences the system performance strongly, especially at low velocity [22]. Here, LuGre dynamic friction model [24] is selected because of its ability to describe the sliding displacement, memorial friction, variable static friction, and viscous friction synchronously. It is given as

\[ T_f = \sigma_0 z + \sigma_1 \frac{dz}{dt} + \sigma_2 \frac{d \theta_m}{dt} \]  

(3)

\[ \frac{dz}{dt} = \dot{z} = \dot{\theta}_m - \frac{\left| \dot{\theta}_m \right|}{g(\theta_m)} \]  

(4)

\[ g(\theta_m) = \frac{1}{\sigma_0} [T_e + (T_s - T_e) e^{-\left| \dot{\theta}_m \right| \theta_m^2}] \]  

(5)

where \( \sigma_0 \) and \( \sigma_1 \) are the dynamic parameters and represents the stiffness coefficient and damping coefficient, respectively. Static parameters are given by \( \sigma_2, T_e, T_s, \) and \( \dot{\theta}_s \), which represent viscous coefficient, Coulomb friction torque, static friction torque, and Stribeck velocity, respectively. \( z \) is the dynamics of the friction internal state.

3 Fuzzy PID Controller

Fuzzy PID (FPID) based on system knowledge is a kind of integrated controller, integration of fuzzy PI and fuzzy PD, to achieve good control performance both in dynamic response and robustness.

Typical structure of FPID is shown in Fig. 2 where error (e) and change error (ce) are the input variables and output (u) is the output variable. Scaling factors S1, S2, and S4 are employed to normalize and map the variables into universe of discourse. S3 is a positive constant for integral gain. The design process can be categorized into following phases, in which the brief illustrations are given as follows:

Fuzzification phase. This phase converts the real value into fuzzy sets. Triangular membership functions are employed to quantify the meanings of linguistic values [25] for the input and output variables, as shown in Figs. 6–8, respectively. These are selected due to their computational efficiency and better control performance [25,26]. Further, nonlinear scaling factors are employed to map the variables into universe of discourse.

Fuzzy rules. These describe the quantitative relationship between variables in linguistic terms [2]. The rule base of fuzzy system is derived using GA, described in Sec. 4.

Inference phase. In this phase, the overall fuzzy control output is computed based on the individual contribution of each rule in the rule base [2]. Direct inference technique is employed for this research study.

Defuzzification phase. In this phase, a real value is produced from the result of inference, which is termed as fuzzy incremental control input. Center of gravity technique is used due to its computational efficiency.

Fuzzy incremental control input is then denormalized using nonlinear scaling factor to obtain the real control input.

4 Tuning of Fuzzy Rule Base

Tuning of FRBSs can be defined into two phases:

1. Learning phase. In this phase, initial rule base are derived automatically from the numeric data using genetic algorithm.

2. Tuning phase. In this phase, the rule base obtained from learning phase are adjusted with slight modifications to increase system performance.

The genetic learning of initial rules is conceded with the following prior information:

• The FRBSs consist of two input (error and change error) and one output (control) variables.

• The input variables can be described into seven linguistic terms uniformly distributed, such that \( e \) and \( ce \) = [NB, NM, NS, ZE, PS, PM, PB].

• There are 49 rules that are based on logical reasoning.

The objective function will be to minimize absolute tracking error.
5 Evolutionary Simultaneous Tuning

In this section, the rule base obtained from learning phase is slightly modified to improve the system performance. Performance of fuzzy PID controller is strongly dependent on the following factors [2,25,26]:

- rule base generation
- nonlinear scaling factors
- membership function

Prior to developing proposed simultaneous tuning process, this section discusses impact of tuning factors on control performance, which is computed using statistical techniques and MINITAB software. The analysis is based on hypothesis test, which is a well known technique of making statistical decisions using experimental data. The value of the $p$-value represents a decreasing index of the reliability of a result, which means that very low $p$-value ($p$-value $< 0.005$) represents very high statistically significant results and vice versa [27,28]. The results are shown in Table 2.

### Table 2 Hypothesis analysis for different factors

<table>
<thead>
<tr>
<th>Distribution</th>
<th>$P$-value</th>
<th>$R^2$-sq</th>
<th>$R^2$-sq (adj)</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scaling factors</td>
<td>$&lt; 0.005$</td>
<td>$&gt; 90$</td>
<td>$&gt; 90$</td>
<td>Reject $H_0$</td>
</tr>
<tr>
<td>Types of MFs</td>
<td>0.779</td>
<td>4.08</td>
<td>0</td>
<td>Accept $H_0$</td>
</tr>
<tr>
<td>MF distribution</td>
<td>$&lt; 0.005$</td>
<td>89.57</td>
<td>87.83</td>
<td>Reject $H_1$</td>
</tr>
<tr>
<td>Null hypothesis:</td>
<td>$H_0$: There is no effect of factors on control performance</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alternate hypothesis: $H_1$: There is strong effect of factors on control performance</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The most common way of tuning MFs is to change their basic parameters, i.e., core and support. Consider triangular MF, we have

$$ u = \begin{cases} 
  x-a & \text{if} \quad a \leq x < b \\
  b-a & \text{if} \quad b \leq x \leq c \\
  0 & \text{otherwise}
\end{cases} $$

(7)

The shape of the fuzzy set associated to the MFs can be altered by changing basic parameters $a$, $b$, and $c$, as shown in Fig. 4, thus influencing the system performance.

The other way to tune MFs is the use of linguistic hedge, which alter the compatibility degree to the fuzzy sets. Two of the well known hedges are the concentration linguistic hedge “very” and the dilation linguistic hedge “more-or-less,” as shown in Fig. 4, which can be described as

$$
\mu_{very}^r(x) = (\mu_r(x))^2
$$

(8)

Aforementioned hypothesis analysis and discussions facilitate to select type and distribution style of MFs. However, proper selection of scaling factors and MF’s distribution are very critical, as very narrower distribution may cause large oscillation and over-shoot [26]. This can be solved by optimizing the required parameters.

Global optimization for complex systems is one of the vital issues and it captivates the attention of researchers. Conventional optimization techniques cannot ensure global optimal state while the contemporary evolutionary algorithm [29–35] could achieve the global optimal state by applying self-adaptation in strategy parameters of the object variables. While optimizing the objective function, the self-adaptation technique requires optimization of the strategy parameters as well. This dual optimization creates algorithmic complexity and in turn, it causes expense in compu-

### Table 1 Fuzzy rules

<table>
<thead>
<tr>
<th>E</th>
<th>NB</th>
<th>NM</th>
<th>NS</th>
<th>ZE</th>
<th>PS</th>
<th>PM</th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td>NL</td>
<td>NL</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NM</td>
<td>NS</td>
</tr>
<tr>
<td>NM</td>
<td>NL</td>
<td>NB</td>
<td>NB</td>
<td>NM</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td>NS</td>
<td>NB</td>
<td>NB</td>
<td>NM</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td>ZE</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>ZE</td>
<td>PS</td>
<td>PS</td>
<td>PS</td>
</tr>
<tr>
<td>PS</td>
<td>PS</td>
<td>PS</td>
<td>PS</td>
<td>PS</td>
<td>PM</td>
<td>PB</td>
<td>PB</td>
</tr>
<tr>
<td>PM</td>
<td>PS</td>
<td>PS</td>
<td>PS</td>
<td>PM</td>
<td>PB</td>
<td>PB</td>
<td>PB</td>
</tr>
<tr>
<td>PB</td>
<td>PS</td>
<td>PS</td>
<td>PS</td>
<td>PM</td>
<td>FB</td>
<td>PB</td>
<td>PL</td>
</tr>
</tbody>
</table>

Initial fuzzy rule generation: 49 Cells

**Fig. 3 Genetic rule base learning for fuzzy PID**
This paper presents the application of novel evolutionary algorithm (NEA), which uses a unique recombination operator, which has a collective interaction of individuals within a particular subpopulation of the population. The mutation strategy parameter is adopted by a time varying function. Furthermore, an elitist selection mechanism is also adopted to make sure that the best individuals are always available to reproduce the new solutions.

5.3 Novel Evolutionary Algorithm. NEA evaluates more than one area of search space and can discover more than one solution to a problem. Typical functions involved are population initialization, fitness evaluation, subpopulation-based max-mean arithmetical crossover (SBMAC), time-variant mutation (TVM), and alternate generation. Typical flow diagram of NEA is shown in Fig. 5.

In this section, NEA is applied to optimize simultaneously non-linear scaling factors, MF distribution, and linguistic hedges for MFs. The object variables can be categorized as follows:

- **Scaling factors.** These are represented by \( S_1, S_2, S_3, S_4 \) to improve the capability of mapping the variables into universe of discourse, as shown in Fig. 2.

- **MF Distribution.** For this core and support, parameters of MFs are selected based on extensive simulation work to improve the accuracy, while ensuring meaningful fuzzy sets. For error and change error MFs, these are represented by \( S_5, S_6, S_7, S_8, S_9 \) and \( S_{10}, S_{11}, S_{12}, S_{13}, S_{14} \), as illustrated in Figs. 6 and 7, respectively. Whereas, \( S_{15}, S_{16}, S_{17}, S_{18}, S_{19}, S_{20} \) represent control output MFs, as illustrated in Fig. 8.

- **Linguistic hedges.** These are represented by \( S_{21}, S_{22}, S_{23} \) for error, change error and control output MFs, respectively. They serve to ensure interpretability avoiding any loss.

The application of novel evolutionary algorithm for optimal FPID works as follows:

1. The initial population \( P(0) \) is selected to consist of \( \mu = 40 \) individuals for each object variable \( S_k, k = 1, \ldots, 23 \) according to the empirical statistics. The initial population is randomly initialized using a uniform random number (URN) within the desired domains of object variables. This initialized population is considered as parent population for next generation after evaluating the individuals \( \mu \) to their cost function.

2. Based on the natural ecosystem, SBMAC concept is used to reduce the genetic variations and make algorithm computationally fast. After ordering the individuals to their cost function, the parent population \( P(t) \) is divided into \( l = 4 \) subpopulations in each generation \( t \), such that each subpopulation has \( \mu / l = 10 \) individuals. The first individual of each object variables of \( i \)th subpopulation is selected as an elite individual \( \psi_{i,\text{max}}(\text{Mom}) \) at generation \( t \), because it maximized a cost function within \( i \)th subpopulation, and a mean individual \( \bar{\psi}_{i,\text{mean}}(\text{Dad}) \) is created from the remaining individuals of \( i \)th subpopulation excluding the \( \psi_{i,\text{max}} \), which is given as:

![Fig. 4 Membership function tuning (a) extended tuning and (b) basic parameters tuning](image)

![Fig. 5 Work flow diagram of NEA](image)

![Fig. 6 Distribution optimization for error MFs](image)

![Fig. 7 Distribution optimization for change error MFs](image)

![Fig. 8 Distribution optimization for control MFS](image)
The crossover operation is defined, to produce two off-springs $(\xi'_1, \xi'_2)$, as

$$\xi'_1 = (\xi_{31}, \xi_{32}, \xi_{33}, \cdots, \xi_{22}, \xi_{23})$$

$$\xi'_2 = \begin{pmatrix} (1 - \alpha_1)S_{11,\text{max}} + \alpha_1 S_{11,\text{mean}} \\
(1 - \alpha_2)S_{21,\text{mean}} \\
\vdots \\
(1 - \alpha_{23})S_{23,\text{mean}} \end{pmatrix}$$

where $S_{1,2,3,\ldots,23}$ represent the object variables to be optimized, $(S_{11,\text{max}}, S_{21,\text{max}}, S_{31,\text{max}}, \ldots, S_{23,\text{max}})$ represent Mom and $(\bar{S}_{11,\text{mean}}, \bar{S}_{21,\text{mean}}, \bar{S}_{31,\text{mean}}, \ldots, \bar{S}_{23,\text{mean}})$ represent Dad, respectively, for object variables as explained above, while $(\xi_{31}, \xi_{32}, \xi_{33}, \cdots, \xi_{22}, \xi_{23})$ represent the corresponding offspring, $\alpha_k, k=1,2,3,\ldots,23$, is selected from URN $[0, 1]$ and $\alpha_k$ is sampled anew for each selected parameter of the individuals.

The crossover operation is defined, to produce two off-springs $(\bar{\xi}_1, \bar{\xi}_2)$, as

$$\bar{\xi}_1 = (\bar{\xi}_{31}, \bar{\xi}_{32}, \bar{\xi}_{33}, \cdots, \bar{\xi}_{22}, \bar{\xi}_{23})$$

$$\bar{\xi}_2 = \begin{pmatrix} (1 - \alpha_1)S_{11,\text{max}} + \alpha_1 S_{11,\text{mean}} \\
(1 - \alpha_2)S_{21,\text{mean}} \\
\vdots \\
(1 - \alpha_{23})S_{23,\text{mean}} \end{pmatrix}$$

(3) The mutation phase provides random excursions into a new location of search space. For this phase dynamic TVM operator is used to improve fine local tuning and to ensure fast convergence. TVM operation for off-springs is defined as

$$\xi''_1 = (\xi_{31}', \xi_{32}', \xi_{33}', \cdots, \xi_{22}', \xi_{23}')$$

$$\xi''_2 = (\xi_{31}' + \sigma(1) \cdot N_{1}(0,1), \xi_{32}' + \sigma(1) \cdot N_{2}(0,1), \cdots, \xi_{22}' + \sigma(1) \cdot N_{23}(0,1))$$

where $\xi_{31}', \xi_{32}', \xi_{33}', \cdots, \xi_{22}'$ represent the new offspring, $N_{1}(0,1)$ indicates the Gaussian random value with zero-mean and standard deviation, and is sampled anew for each value of the index $k$. And $\sigma(t)$ is the time-variant mutation step generating function at the generation $t$, which is defined by

$$\sigma(t) = \left[1 - r^{(1 - \gamma T)^{T}}\right]$$

where $r$ is selected from URN $[0, 1]$, $T$ is the maximal generation number, and $\gamma=6$ is a real-valued parameter determining the degree of dependency on the generations.

(4) After mutation operation, each offspring is evaluated by its fitness function. The modified fitness function is developed to improve tracking performance and reduce any possible chattering effect. The fitness function is defined as

$$J = w_1 \sum_{i=0}^{T_0} |e(t)| + w_2 \sum_{i=0}^{T_0} |ce(t)| + w_3 \sum_{i=0}^{T_0} |u(t)|$$

where $w_1$ is the weighting factor to improve tracking performance, $w_2$ is the weighting factor to reduce steady state error, $w_3$ is the weighting factor to minimize control energy, $T_0$ is the total time for which the function is evaluated, $|e(t)|$ is the absolute error, $|ce(t)|$ is the absolute change error, and $|u(t)|$ is the absolute control output. The lower the value of $J$, the better the performance.

(5) In the alternate generation phase parent $\mu^{t-1}$ and children $\mu^t$ are combined and ordered according to their fitness function. Best $\mu$ individuals are selected for the next generation.

This loop continues till final condition is met and the best solution is obtained.

Optimization curve shown in Fig. 9 confirms the rapid convergence and stable behavior. Table 3 gives the detail of optimized parameters for MFs and scaling factors.

### Table 3 Optimized scaling factors and MF values

<table>
<thead>
<tr>
<th>MFs</th>
<th>Error</th>
<th>Change error</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>NL</td>
<td>-</td>
<td>-</td>
<td>$[-1.25, -0.92, -0.75]$</td>
</tr>
<tr>
<td>NB</td>
<td>$[-1 \times 10^{-7} -0.78 -0.54]$</td>
<td>$[-1 \times 10^{-7} -0.85 -0.55]$</td>
<td>$[-1 -0.64 -0.43]$</td>
</tr>
<tr>
<td>NM</td>
<td>$[-0.78 -0.54 -0.23]$</td>
<td>$[-0.85 -0.55 -0.19]$</td>
<td>$[-0.64 -0.43 -0.25]$</td>
</tr>
<tr>
<td>NS</td>
<td>$[-0.44 -0.165 0]$</td>
<td>$[-0.44 -0.14 0]$</td>
<td>$[-0.43 -0.17 0]$</td>
</tr>
<tr>
<td>ZE</td>
<td>$[-0.165 0.165]$</td>
<td>$[-0.14 0.14]$</td>
<td>$[-0.17 0.17]$</td>
</tr>
<tr>
<td>PS</td>
<td>$[0.165 0.44]$</td>
<td>$[0.14 0.44]$</td>
<td>$[0.17 0.43]$</td>
</tr>
<tr>
<td>PM</td>
<td>$[0.25 0.54 0.78]$</td>
<td>$[0.198 0.55 0.85]$</td>
<td>$[0.25 0.43 0.64]$</td>
</tr>
<tr>
<td>PB</td>
<td>$[0.54 0.78 1 \times 10^{7}]$</td>
<td>$[0.55 0.85 1 \times 10^{7}]$</td>
<td>$[0.43 0.64 1]$</td>
</tr>
<tr>
<td>PL</td>
<td>-</td>
<td>-</td>
<td>$[0.75 0.92 1.25]$</td>
</tr>
</tbody>
</table>
5.4 Convergence Analysis. The performance of an optimization algorithm is not an unambiguous concept, but it depends on the application and implementation. Convergence velocity and convergence reliability are the key performance indicators of an optimization algorithm [21,36,37]. Convergence velocity describes how fast the algorithm is capable of finding the best solution, and NEA, novel evolutionary algorithm aforementioned can ensure the optimization of parameters $S_1, S_2, S_3, \ldots, S_{23}$ with rapid convergence and stable behavior shown in Fig. 9.

To further elaborate the rapid convergence behavior of NEA, a comparative study is carried out between NEA and GA. The objective is to minimize the Bohachevsky function 2 [21], given as

$$f(x_1,x_2) = x_1^2 + 2x_2^2 - 0.3(\cos(3\pi x_1) \cos(4\pi x_2)) + 0.3 \quad (18)$$

The function has a global minimum value of 0 at $(x_1, x_2) = (0, 0)$. The results obtained after $N=25$ trials are plotted in Fig. 10. Threshold refers the acceptable solution, $f(t) = 1e^{-6}$, and the deviation range representing the number of generations for which the algorithm approach of the threshold is statistically computed based on N trials. Optimal solution assumed to be $f(x) < 1e^{-15} = 0$. From Fig. 10, it is evident that NEA converges to the optimal solution rapidly.

6 Rule Set Reduction

At initial stage of fuzzy logic control design and analysis, a large number of fuzzy rules should be used to reach an acceptable accuracy degree. However, an excessive number of rules make no improvement in performance and causes the loss in interpretability of the FRBSs. For large fuzzy rule set, different kind of rules can be categorized as irrelevant rules, which do not contain significant information; redundant rules, whose actions are covered by other rules; erroneous rules, which are ill defined and distort the fuzzy rule base system (FRBS) performance; and conflictive rules, which perturb the FRBS performance when they co-exist with others [3,9,19].

To cope with this problem, a fuzzy rule set reduction technique is developed to achieve the goal of minimizing the number of rules used while maintaining the performance and thus improving the interpretability. The proposed technique involves obtaining an optimized subset of rules from a previous fuzzy rule set by selecting most appropriate. Key steps of the developed technique are explained as follows:

1. Firing strength of each rule under different disturbances is computed using the following relation:

$$F_d R_j = \sum_{t=0}^{T_0} a(t)$$

where $F_d$ is the firing strength of rule $j$ under different operating conditions, $R_j$ is the $j$th rule, $j=1,2,3,\ldots,49$, $T_0$ is the total evaluation time, $a(t)$ is the firing strength at particular instant.

2. Utilizing the statistical approach, range of firing strength for each rule is calculated with a confidence interval of 99.9%. Maximum value is selected to ensure that the corresponding rule must be activated for any additional disturbance effect. Thus

$$\delta R_j = \delta \delta$$

3. Then a reference value is computed, based on selecting the maximum value of all the rules, after doing step 2.

$$\delta_{ref} = \max(\delta R_j)$$

4. Activation tendency of each rule is calculated and categorized the rules into four zones. Rules with zero activation tendencies are treated as dead zone rules, which mean that they will not activate even for additional disturbance. Rules with low tendencies are treated as sleeping zone, which means that they have very low activation tendency under normal operating conditions and their activation tendency increase with additional disturbance effects. Rules with medium tendencies are treated as sitting zone and rules with very high tendencies are treated as active zone. Active zone rules remain active for all the conditions. Table 4 shows the different zones and their specific values. Fuzzy rule set zone distribution is illustrated in Table 5.

<table>
<thead>
<tr>
<th>Zone</th>
<th>No. of rules</th>
<th>Activation tendencies (%)</th>
<th>% zone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dead zone</td>
<td>1–7, 8, 9, 13, 14, 15, 16, 20, 21, 22, 23, 27, 28, 29, 30, 34, 35, 36, 37, 41, 42, 43–49</td>
<td>0</td>
<td>69.38</td>
</tr>
<tr>
<td>Sleeping zone</td>
<td>10, 11, 12, 17, 19, 24, 26, 31, 33, 38, 40</td>
<td>1.38 $\times 10^{-3}$, 1.1 $\times 10^{-1}$, 2.2 $\times 10^{-3}$, 2.17 $\times 10^{-3}$, 3.86 $\times 10^{-4}$, 3.79 $\times 10^{-4}$, 9.18 $\times 10^{-4}$, 9.129 $\times 10^{-4}$, 7.9 $\times 10^{-4}$, 7.98 $\times 10^{-5}$</td>
<td>22.44</td>
</tr>
<tr>
<td>Sitting zone</td>
<td>18, 32, 39</td>
<td>4.689, 2.4685, 0.5</td>
<td>6.12</td>
</tr>
<tr>
<td>Active zone</td>
<td>25</td>
<td>100</td>
<td>2.04</td>
</tr>
</tbody>
</table>

Table 4 Activation tendencies of fuzzy rule base system

<table>
<thead>
<tr>
<th>EC</th>
<th>NB</th>
<th>NM</th>
<th>NS</th>
<th>ZE</th>
<th>PS</th>
<th>PM</th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dead Zone</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NB</td>
<td>Sleeping Zone</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Sitting Zone</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB</td>
<td>Dead Zone</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5 Fuzzy rule set zone distribution
7 Application and Discussion

This section focuses on application of proposed optimal fuzzy PID controller for nonlinear EHSS to achieve precise tracking performance, whose parameters can be found in previous work [36]. Tracking performance before and after optimization is compared in Fig. 11, which elaborates the improvement in performance to much extent.

To further validate the effectiveness of proposed controller and thus the effectiveness of optimization, performance is evaluated under different disturbances. A meticulous comparison of tracking performance is shown in Fig. 12, where nominal (curve 1) represents the performance without any disturbance effect. Very small variations are observed when the friction effect is increased to 50% and supply pressure is changed from $7 \times 10^6$ N/m$^2$ to $8 \times 10^6$ N/m$^2$ as represented by curves 2 and 3, respectively. A constant disturbance torque of 75 N-m is suddenly applied at $t=1$ s, which cause the small change in tracking ($t=1.05–1.1$ s). However this effect is reduced to zero just within 0.05s as illustrated by curve4. Further, disturbance effect is suddenly removed at $t=2$ s; however, negligible effect is observed at $t=2.05$ s, which diminished completely just within 0.05 s. Simulation results presented in Fig. 12 confirms the effectiveness and robustness of proposed controller under all disturbance effects.

The performance of optimized fuzzy PID (OFPID) is then evaluated with reduced rules, i.e., 15 rules, with the same MFs, as given in Table 3, and under the operating conditions as explained above. From simulation results shown in Fig. 13, it is obvious that there is no significant change in performance under different disturbances. Therefore, it is obvious reduced rule set maintains the performance under uncertainties and disturbances.

Further, emphasizing on granularity of partition, the linguistic terms required for error, change error, and control MFs are reduced, as shown in Figs. 14–16, respectively. Consequently, a new reduced rule set is developed consisting of only 15 rules as shown in Table 6.

8 Experimental Validations

In order to validate the effectiveness of the proposed controller, the experiment of EHSS is carried out based on FPID and OFPID with rule base reduction. Experimental setup for EHSS is shown
in Fig. 1. Experimental results shown in Fig. 17 indicate that OFPID with reduced rule base (i.e., 15 rules) maintains the performance as illustrated by curve 3.

To further validate the effectiveness of proposed controller (i.e., OFPID with reduced rule base), performance is evaluated under different disturbances, as shown in Fig. 18. A considerable amount of disturbance is suddenly applied at 2.2 s (curve 2); however, the controller compensates it rapidly and there is no significant change in performance. Further, when the parameter variation and disturbance effect are combined (curve 4), then again the controller has the capability to eliminate this effect rapidly and thus maintains the performance.

Consequently, it might be concluded that the proposed optimization technique and rule base reduction technique not only improve accuracy but also improve interpretability synchronously. Hence, an optimal balance between accuracy and interpretability can be achieved, which improve the system performance.

9 Conclusion

In this contribution, the genetic learning of initial rule base is introduced for linguistic fuzzy modeling. Further, to achieve a reasonable accuracy, the simultaneous tuning of scaling factors and MFs using a novel evolutionary optimization algorithm is developed. The proposed algorithm has the unique recombination operator and adopted mutation strategy parameter, which ensure rapid convergence. To avoid any loss in interpretability, linguistic hedges are given, which slightly modifies the degree of membership. Interpretability issue is further improved by introducing the rule base reduction technique. The proposed technique computes the activation tendencies of each rule and then a significant statistical decision is made using the experimental data. Based on this statistical analysis, most appropriate rules are selected. Further, linguistic terms of variables are redesigned to address the issue of granularity of partition. A new reduced rule base is developed with the objectives to reduce the computational efforts while maintaining the desired performance. The proposed techniques are applied to nonlinear electrohydraulic servo system to ensure precise tracking performance. Extensive simulation and experiment results indicate that proposed schemes not only improve the accuracy but also ensure interpretability preservation. Further, controller developed based on proposed schemes sustains the performance under parametric uncertainties and disturbances. Therefore, it can be concluded that proposed techniques for auto learning of rules, simultaneous optimization and rule set reduction have a lot of potential for reducing the very complex large fuzzy rule base system while maintaining the desired performance.

Table 6 New fuzzy rule set

<table>
<thead>
<tr>
<th>EC</th>
<th>NB</th>
<th>NM</th>
<th>ZE</th>
<th>PM</th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>NB</td>
<td>NM</td>
<td>NS</td>
<td>PS</td>
<td>PM</td>
</tr>
<tr>
<td>ZE</td>
<td>NM</td>
<td>NS</td>
<td>ZE</td>
<td>PS</td>
<td>PM</td>
</tr>
<tr>
<td>P</td>
<td>NM</td>
<td>NS</td>
<td>PS</td>
<td>PM</td>
<td>PB</td>
</tr>
</tbody>
</table>

Table 7 Reduced rule base performance analysis

<table>
<thead>
<tr>
<th>Rule base system</th>
<th>Comparison of tracking error under different effects</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Normal</td>
</tr>
<tr>
<td>Original rule base (49 rules)</td>
<td>499.2710</td>
</tr>
<tr>
<td>Reduce rule base (15 rules)</td>
<td>499.5315</td>
</tr>
<tr>
<td>New reduce rule base (15 rules)</td>
<td>499.5315</td>
</tr>
<tr>
<td>%difference</td>
<td>&lt;0.1</td>
</tr>
</tbody>
</table>

Fig. 17 Experiments results for FPID before and after optimization

Fig. 18 Experiments for new rule base under different effects

References


