Abstract—In this proposed work, Fractional Order PI (FOPI) controller is used to control the pH process using Particle Swarm Optimization algorithm (PSO). The finding of optimal parameters of FOPI controller is originated as a nonlinear optimization problem, in which the objective function is formulated as the combination of Integral Time Absolute Error (ITAE), steady-state error, rise time and settling time. The performances of the PSO-FOPI and PSO-PI controller are compared with Genetic Algorithm (GA) based FOPI and PI controller. Various numerical simulations and comparisons with GA based FOPI/PID controller show that the PSO-FOPI controller can ensure better control performance.

Keywords—FOPI controller; PSO; Genetic Algorithm; pH Process; Performance criterion; ITAE.

I. INTRODUCTION

The Control of pH process is a very challenging task, due to its high process nonlinearity and also plays a very important role in chemical industries such as waste water treatment, polymerization reactions, fatty acid production, biochemical processes, etc. In the recent decade, many modern control techniques including neural network control, fuzzy control, optimal control, predictive control and some hybrid intelligent control have been developed extensively. Nonetheless, the Proportional-Integral-Derivative (PID) control technique has been widely used in many process industries till now. Nowadays, to enhance the performance of PID controller, the concept of fractional calculus has been introduced in the literature. Fractional-Order Proportional-Integral-Derivative (FOPID) controller has been derived from the fractional calculus by Podlubny in 1997 [1, 2].

The Fractional Order Controller is the generalization of non-integer orders of conventional controllers. Many design approaches have been reported in literature for the design of FOPID controller such as frequency domain [3, 4], pole distribution [5], state space design [6], and Quantitative feedback theory [7]. Ziegler–Nichols-type rules tuning of FOPID controllers are reported in literature [8]. Fabrizio and Antonio (2011) proposed a set of tuning rules for integer order and fractional order PID controllers for the First-Order-Plus-Dead-Time (FOPDT) model of the process to minimize the integrated absolute error with a constraint on the maximum sensitivity [9]. Evolutionary algorithms based design approach has been investigated by several authors. Evolutionary algorithm such as Genetic Algorithm (GA), Chaotic Ant Swarm (CAS), Differential Evolution (DE), Chaotic multi-objective optimization, Multi-objective Pareto optimization and some hybrid optimization approaches like, differential evolution and particle swarm optimization are used for the design of FOPID controllers based on the process [10-15]. In this proposed work, the PSO is employed to design FOPI controller for pH process. The FOPI controller has three parameters: proportional gain, integral gain and integral order. The objective function for determining the optimal FOPI controller parameters is formulated as the combination of Integral Time Absolute Error (ITAE), steady-state error, rise time and settling time. The proposed work is simulated with different scenarios and their performances are compared with GA.

The concept of fractional calculus is described in section II. In section III, Particle Swarm Optimization (PSO) algorithm has been explained. Section IV discusses the tuning of FOPI controller using PSO for pH process and numerical results are presented in section V. The conclusion is given in section VI.

II. THE CONCEPT OF FRACTIONAL CALCULUS

Fractional calculus is a branch of mathematical analysis that deals with the possibility of taking real number powers of the differentiation operator and integration operator. To define fractional order differentiation and integration, the generalized differ-integrator \( D_t^{\alpha} \) operator is introduced in
fractional calculus. Where \( m (m \in \mathbb{R}) \) is the fractional order and \( a \) and \( t \) are limits of operation.

There are several definitions of fractional order differentiation and integration. The most commonly used are Riemann and Liouville (RL) and Grunwald–Letnikov (GL) definitions. The GL definition is given by

\[
\frac{d^m f(t)}{dt^{a}} = \lim_{N \to \infty} \left( \frac{(-1)^{N}}{N!} \int_{t-a}^{t} \frac{f(s)}{(t-s)^{N+1}} \, ds \right) \ \ (1)
\]

The RL definition is given by

\[
\frac{d^m f(t)}{dt^{a}} = \frac{1}{\Gamma(n-m)} \frac{d^n}{dt^n} \left( \frac{1}{\Gamma(n)} t^{n-a} f(t) \right) \ \ (2)
\]

for \( n-1 < m < n \) and \( \Gamma(.) \) is the Gamma function.

The Laplace transforms of the RL fractional differ-integral operation for zero initial conditions are given as follows

\[
\mathcal{L}\left\{ \frac{d^m f(t)}{dt^{a}} \right\} = s^m F(s) \ \ (3)
\]

In practice, fractional-order differential equations do not have exact analytic solutions. Hence, the transfer functions involving fractional orders of \( s \) are approximated with typical (integer order) transfer functions. Several possible approximation methods are available in literature [3]. One of the well known approximations is Oustaloup which uses the recursive distribution of poles and zeros and is given by [16]

\[
s^m = k \prod_{n=1}^{N} \left( 1 + \left( \frac{s}{\omega_m} \right)^m \right), \quad m > 0 \ \ (4)
\]

The approximation is valid in the frequency range \([\omega_h, \omega_b]\), gain \( k \) is adjusted so that, the approximation shall have a unit gain at 1 rad/sec, the number of poles and zeros \( N \) is chosen beforehand. The best performance of approximation strongly depends on \( N \), where as low values result in simpler approximations but also cause the appearance of a ripple in both gain and phase behaviors. Such ripples can be functionally removed by increasing \( N \), but approximation becomes computationally heavier. Frequencies of poles and zeros in Eqn. (4) are given by

\[
\alpha = \left( \frac{\omega_h}{\omega_b} \right)^{m} \ , \ \ (5)
\]

\[
\eta = \left( \frac{\omega_h}{\omega_b} \right)^{(1-m)} \ \ (6)
\]

\[
\omega_{\alpha} = \omega_1 \sqrt{\eta} \ \ (7)
\]

\[
\omega_{\eta} = \omega_{\eta, n-1}, \quad n = 2, ..., N \ \ (8)
\]

\[
\omega_{\alpha} = \omega_{\alpha, n-1} \ , \quad n = 1, ..., N \ \ (9)
\]

The case \( m < 0 \) can be dealt with inverting (4). For \( m > 1 \), the approximation becomes dissatisfied. So, it is common to separate the fractional order of \( s \) as follows

\[
s^\delta = s^{n+\delta}, \quad m = n + \delta, \quad n \in \mathbb{Z}, \quad \delta \in [0, 1] \ \ (10)
\]

As a result, only the second term \( (s^\delta) \) has to be approximated.

III. PARTICLE SWARM OPTIMIZATION (PSO)

Particle Swarm Optimization (PSO) algorithm is a population based stochastic technique inspired by social behaviour of bird flocking or fish schooling [17]. Nevertheless, unlike GA, PSO has no evolution operators such as crossover and mutation. Compared to GA, it is easy to implement and there are few parameters to adjust. The PSO consists of a swarm of particles moving in a D dimensional search space where certain quality measures and fitness are being optimized. The position and velocity of each particle are represented by a position vector \( \mathbf{X}_i \) \((x_{i1}, x_{i2}, ..., x_{iD})\) and velocity vector \( \mathbf{V}_i \) \((v_{i1}, v_{i2}, ..., v_{iD})\), respectively. Each particle remembers its own best position in a vector \( \mathbf{P}_i \) \((p_{i1}, p_{i2}, ..., p_{iD})\). Where \( i \) is the index of that particle. The best position vector among all the neighbours of a particle is then stored in the particle as a vector \( \mathbf{P}_g \) \((p_{g1}, p_{g2}, ..., p_{gD})\). The velocity and position of each particle can be manipulated according to the following equations.

\[
\mathbf{V}^{(i+1)} = \mathbf{wV}^{(i)} + \mathbf{c}_1 \mathbf{r}_1 (\mathbf{p}^{(i)} - \mathbf{x}^{(i)}) + \mathbf{c}_2 \mathbf{r}_2 (\mathbf{p}_g^{(i)} - \mathbf{x}^{(i)}) \ \ (11)
\]

\[
\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \mathbf{V}^{(i+1)}, \quad m = 1, 2, ..., D \ \ (12)
\]

Where, \( w \) is inertia weight and also decreases linearly from 0.9 to 0.4 during a run. In this study, inertia weight (w) is set to 0.9 [18]. Also \( \mathbf{c}_1 \) and \( \mathbf{c}_2 \) are the acceleration constants which influence the convergence speed of each particle and are often set to 2.0, according to the past experiences [19]. In addition to that, \( r_1 \) and \( r_2 \) are random numbers in the range of [0, 1].

IV. TUNING OF FOPID CONTROLLER USING PSO FOR PI\textsuperscript{H} PROCESS

A. FOPID Controller

The FOPID controller is a generalization of integer order PID controller, which can be called as \( \text{PI}^\delta \text{D}^\delta \) controller. The continuous transfer function of such a controller is given by

\[
G_c(s) = K_p + K_i s^{-\delta} + K_d s^\delta, \quad (\lambda, \mu > 0) \ \ (13)
\]

The time domain equation for the FOPID controller output is given by

\[
u(t) = K_p e(t) + K_i D^{-\delta} e(t) + K_d D^{\delta} e(t) \ \ (14)
\]

From Fig.1, it can be seen that FOPID controller generalizes the integer order PID controller and expands it from point to plane. This expansion can give more flexibility to the design of controller in achieving control objectives.
The integer order PID controller is obtained when \( \lambda = \mu = 1 \). If \( \lambda = 1 \) and \( \mu = 0 \), normal PI controller is obtained. In this proposed work, FOPI controller is employed for the control of pH process. The output equation and continuous transfer function of FOPI controller are obtained from the Eqsns. 16 and 17 as follows:

\[
u_i(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (15)
\]

\[
G_c(s) = K_p + K_i s^{-\lambda} + K_d s^{-\mu}, \quad (\lambda > 0)
\quad (16)
\]

**B. pH process modelling**

Heads, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

\[ \text{pH} = -\log_{10}[H^+] \quad (17) \]

The pH process consists of neutralization of two monoprotic reagents of a weak acid (acetic acid) and a strong base (sodium hydroxide). The method uses mass balances on components called reaction invariants of the solution in the Continuous Stirred Tank Reactor (CSTR) as shown in Fig.2. The CSTR has two inlet streams: the influent process stream, the titrating stream and one effluent stream at the output. The model of the pH neutralization process used in this work follows the method proposed by McAvoy et al. (1972) and is given below [20]. Assumption of perfect mixing is general in the modeling of pH processes. Material balances in the reactor can be given by

\[
\frac{dx_a}{dt} = F_a C_a - (F_a + F_b) x_a \quad (18)
\]

\[
\frac{dx_b}{dt} = F_b C_b - (F_a + F_b) x_b \quad (19)
\]

Where \( F_a \) is the flow rate of the influent stream, \( F_b \) is the flow rate of the titrating stream, \( C_a \) is the concentration of the influent stream, \( C_b \) is the concentration of the titrating stream, \( x_a \) is the concentration of the acid solution, \( x_b \) is the concentration of the basic solution and \( V \) is the volume of the mixture in the CSTR.

The Eqsns. 18 and 19 describe the concentration of the acidic and basic components \( x_a \) and \( x_b \) which change dynamically subject to the input streams \( F_a \) and \( F_b \). Consider acetic acid (weak acid) denoted by HAC and is neutralized by a strong base NaOH (sodium hydroxide) in water. The reactions are

\[
H_2O \Leftrightarrow H^+ + OH^- \quad (20)
\]

\[
HAC \Leftrightarrow H^+ + AC^- \quad (21)
\]

\[
NaOH \Leftrightarrow Na^+ + OH^- \quad (22)
\]

According to the electric neutrality condition, the sum of the charges of all ions in the solution must be zero, i.e.

\[
[Na^+] + [H^+] = [AC^-] + [OH^-] \quad (23)
\]

Here, \([X] \) denotes the concentration of the X ion. The equilibrium relations also hold water and the acetic acid.

\[
K_a = \frac{[AC^-] [H^+]}{[HAC]} \quad (24)
\]

defining, \( X_a = [HAC] + [AC^-] \) and \( X_b = [Na^+] \) using the Eqsns. (23) and (24)

\[
[H^+]^2 + [H^+] [K_a + X_b] + [H^+] [K_a (X_b - X_a) - K_a] - K_a K_a = 0 \quad (25)
\]

Let \( pK_a = -\log_{10} K_a \) and \( \rho K_a = -\log_{10} K_a \).

The equation for titration is given by

\[
X_a + 10^{-\rho pH} - 10^{-\rho pH + \rho K_a} - \frac{X_a}{1 + 10^{\rho K_a - \rho pH}} = 0 \quad (26)
\]

Where, \( K_a \) and \( K_a \) are dissociation constant of acetic acid at 25°C \( (K_a = 1.778 \times 10^{-5} \) & \( K_a = 10^{-14}) \). Fig.3 shows the titration curve drawn between acid / base and pH using the Eqn. 26. It is strictly static nonlinear relation between the states \( X_a, X_b \) and the output pH variable, and it manifests itself as the familiar titration curve of the neutralization process.
In this work, a new performance criterion in the time domain is proposed for design of FOPI controller for the control of pH process. This performance criterion consists of ITAE, rise time (t_r) , settling time (t_s) and steady-state error (E_s). The ITAE has to compute numerically and the time limit for the integral is set from 0 to T which is chosen sufficiently as large with the intention that, error is negligible for t is greater than T. The proposed performance criterion J(k) is defined as

\[ J(k) = W_1 \times \text{ITAE} + W_2 \times E_s + W_3 \times t_r + W_4 \times t_s \]  

(28)

Where, k is the parameter of FOPI controller \( \{k_p, k_i, \lambda\} \)

The importance of each criterion in Eqn.31 is determined by a weight factor \( W_i \). The user can set the weight factors properly in order to attain the desired requirements. In this study, weight factors selections are \( W_i=1000 \), \( W_j=100 \) and \( W_k=W_\lambda=1 \). A change in \( W_i \) gives improvement in the corresponding characteristics at the expense of degrading other characteristics.

\[ J(k) = \int_0^\infty t |e(t)| dt \]  

(27)

In this work, a new performance criterion in the time domain is proposed for design of FOPI controller for the control of pH process. This performance criterion consists of ITAE, rise time (t_r) , settling time (t_s) and steady-state error (E_s). The ITAE has to compute numerically and the time limit for the integral is set from 0 to T which is chosen sufficiently as large with the intention that, error is negligible for t is greater than T. The proposed performance criterion J(k) is defined as

\[ J(k) = W_1 \times \text{ITAE} + W_2 \times E_s + W_3 \times t_r + W_4 \times t_s \]  

(28)

Where, k is the parameter of FOPI controller \( \{k_p, k_i, \lambda\} \)

The importance of each criterion in Eqn.31 is determined by a weight factor \( W_i \). The user can set the weight factors properly in order to attain the desired requirements. In this study, weight factors selections are \( W_i=1000 \), \( W_j=100 \) and \( W_k=W_\lambda=1 \). A change in \( W_i \) gives improvement in the corresponding characteristics at the expense of degrading other characteristics.

**D. FOPI Controller design using PSO**

The PSO algorithm is employed to determine the optimal parameters \( \{k_p, k_i, \lambda\} \) of FOPI controller by optimizing a proposed performance criterion. Here, FOPI controller is used to improve the transient step response of pH process. Initially, objective function is formulated as minimization of J(k) as mentioned in Eqn. 30. To make sure the stability of closed loop, the objective function is penalized with a penalty function M(k) and is given by

\[ M(k) = \begin{cases} Q & \text{if k is unstable} \\ 0 & \text{otherwise} \end{cases} \]  

(29)

where, Q is a large positive real number.

Therefore, fitness function for this optimization problem is obtained as

\[ f(k) = J(k) + M(k) \]  

(30)

The design steps of PSO-FOPI controller for pH process are as follows.

1. Randomly initialize the population (position points and velocities)
2. Calculate the values of the performance criterion in Eqn.30 for each initial particle k of the population.
3. Compare each particle’s evaluation value with its personal best Pi. The best evaluation value among the Pi is denoted as Pg.
4. Modify the member velocity of each particle k according to Eqn. 11.
5. Modify the member position of each particle k according to Eqn. 12.
6. If the number of iterations reaches the maximum, then go to Step7, otherwise go to Step2.
7. The latest Pg is the optimal controller parameter.

**V. NUMERICAL RESULTS**

**A. Related Parameters**

The model parameters of the pH process are listed in Table I. In this study, acid flow rate \( (F_a) \) is kept constant at 0.192 l - min \(^{-1}\) and base flow rate \( (F_b) \) is the manipulated variable for the control of pH. The lower and upper bounds of the each FOPI controller parameter are \( 0 \leq k_p, k_i \leq 50 \) and \( 0 \leq \lambda \leq 2 \). In
Oustaloup approximation, \( \omega_l = 0.001 \omega_{gc} \) and \( \omega_h = 1000 \omega_{gc} \), where, \( \omega_{gc} \) is the gain cross over frequency and the order of approximation (N) is set to 5. The block diagram and MATLAB – Simulink diagram of pH process model with FOPI controller are shown in Figs.4 and 5 respectively.

**TABLE I. MODEL PARAMETERS FOR THE pH PROCESS**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V )</td>
<td>Volume of the Continuous Stirred Tank Reactor</td>
<td>7.4l litre</td>
</tr>
<tr>
<td>( F_a )</td>
<td>Flow rate of the influent stream</td>
<td>0.24 l min(^{-1})</td>
</tr>
<tr>
<td>( F_t )</td>
<td>Flow rate of the titrating stream</td>
<td>0.8 l min(^{-1})</td>
</tr>
<tr>
<td>( C_a )</td>
<td>Concentration of the influent stream</td>
<td>0.2 g mol(^{-1})</td>
</tr>
<tr>
<td>( C_t )</td>
<td>Concentration of the titrating stream</td>
<td>0.1 g mol(^{-1})</td>
</tr>
</tbody>
</table>

**Fig. 5. MATLAB- Simulink diagram of pH process with FOPI controller**

**B. Performance of PSO-FOPI Controller**

The performance of PSO-FOPI controller is also analyzed in this section. The performance of proposed PSO-FOPI controller is compared with GA based FOPI controller. The parameters for the GA training are as follows: number of generations = 30, population size = 10, crossover probability = 0.8 and mutation probability = 0.08. The PSO parameters are selected as follows: dimension (d) =3, population size=50, maximum number of bird step = 5, cognitive factor (C\(_1\)) = 1.2, social acceleration factor (C\(_2\)) = 1.2, inertia weight factor (w) \( = 0.9 \), maximum time limit for integral is set to \( T=4 \) sec and Q in Eqn. 29 is set to \( 10^3 \). The GA and PSO algorithm are executed for ten runs to get best controller parameters using the performance criterion mentioned in Eqn.30.

**TABLE II. PERFORMANCE MEASURES AND BEST CONTROLLER PARAMETERS OF VARIOUS PI CONTROLLERS**

<table>
<thead>
<tr>
<th>Set point</th>
<th>Controller</th>
<th>( K_p )</th>
<th>( K_i )</th>
<th>( \lambda )</th>
<th>ITAE</th>
<th>( t_r )</th>
<th>( t_s )</th>
<th>( Ess )</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>GA-FOPI</td>
<td>7.283</td>
<td>26.00</td>
<td>1.79</td>
<td>13.63</td>
<td>0.033</td>
<td>0.221</td>
<td>0.211</td>
</tr>
<tr>
<td></td>
<td>PSO-FOPI</td>
<td>30.108</td>
<td>33.83</td>
<td>0.44</td>
<td>0.89</td>
<td>0.054</td>
<td>0.171</td>
<td>0.029</td>
</tr>
<tr>
<td>9</td>
<td>GA-FOPI</td>
<td>14.467</td>
<td>43.05</td>
<td>0.01</td>
<td>1.58</td>
<td>0.279</td>
<td>0.497</td>
<td>0.009</td>
</tr>
<tr>
<td></td>
<td>PSO-FOPI</td>
<td>29.368</td>
<td>25.12</td>
<td>1.03</td>
<td>1.42</td>
<td>0.196</td>
<td>0.414</td>
<td>0.018</td>
</tr>
</tbody>
</table>

**Fig. 6. Step response of pH process controlled by PSO-FOPI and GA-FOPI Controller**

Set point of pH =7

**Fig. 7. Step response of pH process controlled by PSO-FOPI and GA-FOPI Controller**

Set point of pH =9

**C. Comparison with other PI controller**

The performance of PSO-FOPI controller is also compared with GA-PI and PSO-PI controllers. The step responses of pH process controlled by GA-PI, PSO-PI and PSO-FOPI controller for the set point of pH 7 and 9 are shown in Figs. 8 and 9. The performance measures and the best controller parameters in the time domain are listed in Table III. As shown in Figs. 8, 9 and Table III, the performance of PSO-FOPI controller has better performance with smaller ITAE, rise time, settling time and steady state error compared to GA-PI and PSO-PI controller for both the set points.

**TABLE III. PERFORMANCE MEASURES AND BEST CONTROLLER PARAMETERS OF VARIOUS PI CONTROLLERS WITH PSO-FOPI CONTROLLER**

<table>
<thead>
<tr>
<th>Set point</th>
<th>Controller</th>
<th>( K_p )</th>
<th>( K_i )</th>
<th>( \lambda )</th>
<th>ITAE</th>
<th>( t_r )</th>
<th>( t_s )</th>
<th>( Ess )</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>GA-PI</td>
<td>25.318</td>
<td>33.38</td>
<td>-</td>
<td>14.51</td>
<td>0.063</td>
<td>0.523</td>
<td>0.187</td>
</tr>
<tr>
<td></td>
<td>PSO-PI</td>
<td>23.192</td>
<td>35.83</td>
<td>-</td>
<td>0.978</td>
<td>0.068</td>
<td>0.167</td>
<td>0.022</td>
</tr>
<tr>
<td>9</td>
<td>GA-PI</td>
<td>10.899</td>
<td>5.76</td>
<td>-</td>
<td>4.43</td>
<td>0.344</td>
<td>2.062</td>
<td>0.041</td>
</tr>
</tbody>
</table>
D. Robustness

Finally, robustness of PSO-FOPI controller is demonstrated with the disturbance rejection problem for set point 7 and 9. The task of FOPI controller is to maintain a constant pH value of 7 and 9, when some unmeasured input disturbances act on the acid input stream. To maintain a constant pH value 7, the first input unmeasured disturbance is created by increasing the acid flow rate by 1 l-min\(^{-1}\) from 1.26s to 1.33s. This is equivalent to 5 times of normal acid flow rate. The second disturbance is done by reducing the acid flow rate by 1 l-min\(^{-1}\) from 1.98s to 2.08s. Then to maintain a constant pH value 9, the first input unmeasured disturbance is created by increasing the acid flow rate by 3 l-min\(^{-1}\) from 1.26s to 1.33s. This is equivalent to 15 times of normal acid flow rate. The second disturbance is done by reducing the acid flow rate by 3 l-min\(^{-1}\) from 1.98s to 2.08s. From Figs. 10 and 11, it can be seen that the PSO-FOPI controller reacts quickly with a change of the base stream compared to PSO-PI controller while a deviation in the pH value arises. Moreover, it can be observed that the disturbance does not change the pH value outlying from the set point. The PSO-FOPI controller is more robust to disturbances than the PSO-PI controller.

VI. CONCLUSIONS

In this work, a design method for tuning of FOPI controller parameters using PSO is presented for the control of pH process. The proposed method determines the optimal parameters of FOPI controller by solving the optimization problem for minimizing the objective function comprising ITAE, rise time, settling time and steady state error. The graphical and numerical results of simulation show that the designed PSO-FOPI controller has better performance than GA based FOPI and PI controllers. In addition, from Fig. 8 and 9, it can be concluded that PSO-FOPI controller is more robust to input unmeasured disturbances.

References


