ARTIFICIAL RAINDROPALGORITHM BASED CONTROL FOR FAULT TOLERANCE AND RECONFIGURATION OF DISTILLATION COLUMN

GANESH RM\textsuperscript{a} 
Nandha Engineering College, Erode, 
ramganesh024@gmail.com

Dr. MANOHARAN S\textsuperscript{b} 
Karpagam College of Engineering, Coimbatore 
manoish07@gmail.com

Abstract: In this paper, a novel nature-based meta-heuristic method, named artificial raindrop algorithm is proposed. This algorithm is stimulated from the phenomenon of natural rainfall and applied for nonlinear distillation column using parallel cascade control. Distillation column is an important process in chemical engineering. Distillation is a technique of separating mixtures with given contrasts in volatilities of components in a boiling fluid mixture. Cascade control scheme is efficient for systems that have large time constants and disturbances. It provides fast recovery from disturbances. Among the two methods of cascade such as series and parallel cascade, later reduces steady state error compared to series cascade. In this paper, various controllers such as PID and fuzzy gain scheduling PID controller are analyzed for flow control with fault tolerance in distillation column. Proposed artificial raindrop algorithm based results are compared with the PID and fuzzy gain scheduling PID controller based control to validate the enhanced performance of the proposed system. Entire system is analysed using Matlab/Simulink.

Keywords: Parallel Cascade Controller, Distillation Column, Fault Tolerance, PID, Fuzzy Gain Scheduling Controller, Artificial Raindrop Algorithm.

1. Introduction

In a chemical and petrochemical industry, distillation plays vital role in separation\cite{1-2}. Design and control of distillation systems are important in the aspect of security, pollution and cost reduction\cite{3-5}. In this paper, wood and berry model of distillation column is considered for analysis because the model was established as a good, widely used during recent decades in many researches. Cascade control is extensively used in process industries to lower the effects of probable disturbances and to enhance the dynamic performance of the closed-loop system\cite{6-10}. The cascade control structure consists of two loops: primary (outer) loop and secondary (inner) loop. Cascade control can be classified as series and parallel cascade control. Series cascade is applicable for the system where disturbance and the manipulated variable affect the primary output through the secondary output\cite{11}. Meanwhile, in distillation process the disturbance and manipulated variables concurrently influence primary and secondary outputs, parallel cascade control is proposed in this paper.

Luyben introduced parallel cascade control for distillation column with the comparative analysis of series cascade control\cite{12}. From the analysis, it is noted that parallel cascade control reduces steady state error. PinakPani Biswas et al. (2009) analyzed nonlinear control of distillation column using feedback linearizing systems\cite{13}. A simple nonlinear controller is analyzed by Chien et al. (1997) to control the product quality of distillation columns using model predictive control with disturbance rejection\cite{14}. Daniele Semino and Alessandro Brambilla (1996)\cite{8} analysed various possible Internal Model Control (IMC) structures for distillation control in series and parallel cascade control. The system is analysed for both in the nominal case and in the presence of uncertainties. Puneet Mishra et.al (2015) \cite{15} analysed an intelligent control algorithm of Fractional Order Fuzzy PID controller for distillation control. Simulations results are validated with the comparative analysis of fuzzy PID controller. Lee et al. (2006) proposed an analytical method of PID controller design for parallel cascade control taking into account the interaction between primary and secondary control loops\cite{16}. Conventional control techniques in distillation column control use model of the process which is highly nonlinear and complex. In this article, fuzzy gain scheduling (FGS) PID controller is applied in distillation column control which deals with unknown data.

Novel optimization method of Artificial Rain Drop (ARD) algorithm is proposed in this paper for flow control in distillation. The name of algorithm clearly states that it mimics the varying process of a raindrop, including the generation, descent, collision, flowing and the updating of a raindrop.
2. Parallel Cascade Controller On Distillation Column

Mathematical model of wood and berry distillation column is stated as follows.

\[
\begin{bmatrix} X_D(s) \\ X_B(s) \end{bmatrix} = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} \begin{bmatrix} R(s) \\ S(s) \end{bmatrix}
\]

Where,

- \(X_D\) = Overhead product Composition
- \(X_B\) = Bottom product Composition
- \(P_{11}\) = Direct process transfer function, relating overhead composition to reflux flow
- \(P_{12}\) = Interacting process transfer function, relating overhead composition to steam flow
- \(P_{21}\) = Interacting process transfer function, relating overhead composition to steam flow
- \(P_{22}\) = Direct process transfer function, relating bottoms composition to steam flow
- \(R\) = Reflux flow rate
- \(S\) = Steam flow rate

In a distillation column control of an overhead composition, cascaded onto the control of a tray temperature is a classic model of a parallel cascade control system [18-19]. In case of a cascade control, the reflux flow rate is assigned as manipulated variable and composition or feed flow is considered as disturbance (d). The control of the system decides the purity of the overhead product and tray temperature. Fig 1 shows the parallel cascade control structure in which \(G_p\) and \(G_s\) are the primary and the secondary controller.

![Fig.1. Parallel cascade control](image)

In figure 1, \(P\) is the process transfer function, \(P_d\) is the process disturbance and \(G_c\) is the controller used in cascade control, where 1 and 2 stands for an outer and inner loop. Initially, the system is analyzed with conventional PID controller, then analyzed with FGS-PID and proposed ARD algorithm.

3. Fault Tolerance with Parallel Cascade Control

Fault tolerance is the major concern in the design of process control systems. In the absence of faults, a process will remain at the steady state, and the control is not necessary. Several control algorithms were proposed to improve fault tolerance capability. Two controllers can be designed separately for servo purposes and Fault Tolerance. This kind of approach raises the complication of control system design be designing two controllers for a single pair of input and output. Using the combination of primary and secondary measurements, a simpler approach can be taken to overcome the load fault problems. For this principle, the parallel cascade control structure is used, wherein the primary controller is intended for the servo purpose and the secondary controller is deliberate for the Fault tolerance. In the further analysis, fault is taken as a disturbance in a system. The major criteria for designing reconfiguration controller are its ability to sustain the acceptable performance even after fault. In this paper gain scheduling and optimization controllers are analysed.

Transfer function of the outer loop and inner loop of the system analyzed are given by [16]

\[
P_1 = P_{d1} = \frac{e^{-ds}}{20s + 1} \quad (1)
\]

\[
P_2 = P_{d2} = \frac{1}{10s + 1} \quad (2)
\]

\(P_1\) and \(P_2\) are the process transfer functions of outer and inner loop whereas \(P_{d1}\) and \(P_{d2}\) are disturbance transfer function of the outer loop and inner loop. Simulation analyses are done using Matlab/Simulink and the simulation model is shown in figure 2.
Fig. 2. Simulink model of parallel cascade control

4. PID Controller in Distillation

A Proportional–Integral–Derivative controller (PID controller) is a conventional controller widely used in many industries for its simple control [19-21]. In this paper, Ziegler Nichols method of tuning is used. The Ziegler–Nichols tuning method is a heuristic method of tuning a PID controller. This tuning rule is meant to give PID loops best disturbance rejection. The transfer function of a standard PID [22-23] controller in an ideal form is

$$G(s) = K_p + K_i \left(\frac{1}{s}\right) + K_d s \quad (3)$$

Where $K_p$, $K_i$, and $K_d$ are the proportional, integral and derivative gain respectively. Each gain has different effect on the speed of response and stability. Two PID controllers are proposed in the parallel cascade control to control flow and temperature. Both PID controllers are tuned using Ziegler Nichols method of tuning.

5. Fuzzy Gain Scheduling PID in Distillation

Fuzzy gain scheduling PID is a method of online tuning of gains of PID controller based on control parameters [24-26]. Fixed value of PID controller gains irrespective of change in input oscillates and delays the settling of output [27-30]. Hence in this analysis fuzzy logic controller is proposed to tune the gains of PID controller. In the case of distillation cascade control, it receives flow error as input to produce $K_p$, $K_i$, and $K_d$ as output. In this analysis, FGS PID controller is applied in outer loop and PID controller is applied in inner loop.

In this paper, Mamdani type of fuzzy with the min-max method of fuzzification is used. Centroid method of defuzzification is applied to attain output [31]. Block diagram of FGS PID for distillation control is shown in figure 3.

Rules in the FGS PID controller decides the output based on input whereas the number of rules in controller decides the processing time in real time implementation. In this paper, 25 rules are applied after many analyses. Flow error ($e$) and change in error ($ce$) are inputs of controller with the membership functions of Positive High, Positive low, zero, Negative low and negative high are stated as {PH, PL, Z, NL, NH}. $K_p$, $K_i$ and $K_d$ are outputs with the membership functions of {Z, S, M, B}. Simulation model of FGS-PID is shown in figure 4.

From the figure 4 model, it is noted that flow controller output ($T^*$) of FGS-PID is a reference for temperature controller (inner loop).

Figure 5 shows the membership function of inputs $e$ and $ce$. Figure 6 (a), (b), (c) shows the membership functions of outputs.
The rules are framed such as:

If (E is PH) and (EC is PH) then (K_p is B) (K_i is B) (K_d is Z)

If (E is Z) and (EC is Z) then (K_p is Z) (K_i is Z) (K_d is Z)

If (E is NH) and (EC is PH) then (K_p is M) (K_i is Z) (K_d is Z)

Similarly based on the value of present error and change in error K_p, K_i and K_d are tuned. The table 1, 2, 3 shows the rules for K_p, K_i and K_d.

### Table 1 Rules for K_p

<table>
<thead>
<tr>
<th>ece</th>
<th>NH</th>
<th>NL</th>
<th>Z</th>
<th>PL</th>
<th>PH</th>
</tr>
</thead>
<tbody>
<tr>
<td>NH</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>M</td>
</tr>
<tr>
<td>NS</td>
<td>M</td>
<td>B</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>Z</td>
<td>M</td>
<td>B</td>
<td>Z</td>
<td>S</td>
<td>B</td>
</tr>
<tr>
<td>PL</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>PH</td>
<td>M</td>
<td>B</td>
<td>B</td>
<td>M</td>
<td>B</td>
</tr>
</tbody>
</table>

### Table 2 Rules for K_i

<table>
<thead>
<tr>
<th>ec</th>
<th>NH</th>
<th>NL</th>
<th>Z</th>
<th>PL</th>
<th>PH</th>
</tr>
</thead>
<tbody>
<tr>
<td>NH</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
</tr>
<tr>
<td>NL</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>Z</td>
<td>B</td>
<td>B</td>
<td>Z</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>PL</td>
<td>S</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>PH</td>
<td>Z</td>
<td>S</td>
<td>B</td>
<td>B</td>
<td>B</td>
</tr>
</tbody>
</table>

### Table 3 Rules for K_d


Fuzzy Gain Scheduling controller reduces the overshoot and steady state error, even though the overshoot and steady state error are to be reduced further.

6. Artificial Raindrop Algorithm

From the name of artificial raindrop algorithm, it is clear that it is the inspiration of principle of natural raindrops. This algorithm begins with vapors as initial population. Then, an individual altitude is considered as the individual fitness value. Due to the action of gravity from high altitude, the small raindrops will flow to low altitude direction. At last most of the small raindrops will stay in the location of the optimal solution. The five correspondences of ARD algorithm are discussed below.

Raindrop generation operator: In this step, by absorbing ambient water vapour in nature, the raindrop is generated. The position of the raindrop is assumed that it is the geometric centre of ambient water vapour. At each generation $t$, the raindrop generation operator $\phi^G$ on vapour population.

\[
\text{Pop}(t) = \{\text{Vapor}_1(t), \text{Vapor}_2(t), ..., \text{Vapor}_N(t)\}
\]

is carried out as follows.

Define

\[
\text{Raindrop} = \{\text{Vapor} = \left\{ \sum_{j=1}^{N} \right. \}
\]

where $N$ is the number of population size.

Raindrop descent operator: By the action of gravity the raindrop straight drops from the cloud to the ground in the absence of effect of wind. This is equivalent to a disturbance in an evolutionary algorithm or one-dimensional mutation operator.

Hence, the raindrop descent operator $\phi^D$ on raindrop is executed as follows.

\[
\text{New Raindrop}(t) = \phi^D_k((\text{Pop}(t))) \quad (6)
\]

i.e.

\[
\text{New Raindrop}_{r1}(t) = \text{New Raindrop}_{r2}(t) + \phi \cdot (\text{New Raindrop}_{r1}(t))
\]

where $r_1, r_2, r_3 \in \{1, 2, ..., D\}$ are randomly chosen indexes, $j$ is the corresponding index of decision variable in New Raindrop, and $\phi$ is a random number in the range $(-1, 1)$.

Raindrop collision operator: In order to keep the population scale stability, number of small raindrops produced from the main drop when contacts with the ground are equal to the population size.

\[
\text{Small Raindrop}(t) = \phi^F_k \left( \text{New Raindrop}(t) \cup (\text{Pop}(t)) \right)
\]

i.e.

\[
\text{Small Raindrop}_{ij}(t) = \text{New Raindrop}_{ij}(t) + \text{sign}(\alpha_j) \cdot 0.5
\]

Where $i (i = 1, 2, ..., N)$ and $j (j = 1, 2, ..., D)$ are the index of ith and the corresponding dimension of small raindrop, respectively. $k, \alpha_j, \beta_j$ are two uniformly distributed random
numbers in the range (0, 1) and \( \text{sign}( ) \) stands for sign function. 

**Raindrop flowing operator:** Due to the action of gravity, finally most of the small raindrops will stay in the locations with a relatively low elevation. Raindrop pool (RP) which comprises optimal solutions. The raindrop flowing operator \( \phi_R^F \) on small raindrops is generated as follows.

\[
\text{New Small Raindrop}(t) = \phi_R^F(\text{Small Raindrop}(t)) \tag{10}
\]

i.e. 

\[
\text{New Small Raindrop}_1(t) = \text{Small Raindrop}_1(t) + d(t - \lambda). 
\]

Where 

\[
d(t - \lambda) = \tau_1 \cdot \text{rand}_1 \cdot \text{sign}(F(\text{RP}_k) - F(\text{Small Raindrop}_1(t))) + \tau_2 \cdot \text{rand}_2 \cdot \text{sign}(F(\text{RP}_k) - F(\text{Small Raindrop}_1(t)))
\]

where, in raindrop pool RP, RP\(_1\) and RP\(_2\) are any two of candidate solutions, \( \text{rand}_1 \) and \( \text{rand}_2 \) are two uniformly distributed random numbers in the range (0, 1), \( \tau_1 \) and \( \tau_2 \) are two step parameters of Small Raindrop flowing, \( F( ) \) is the fitness function, \( \text{sign}( ) \) stands for sign function, \( \lambda \) is a damping coefficient and can be chosen by the tournament selection procedure. However, each raindrop could not have been in the flowing in a real environment. They will stay in the locations with a relatively lower elevation or evaporate after numerous flowing. It is essential to commence a parameter \text{Max Flow Number} to manage the maximum number of flowing.

**Raindrop updating operator:** By the evaporation, the water vapor produced in the land water cycle system is mainly from surface water. In order to enhance the convergence rate and computational performance for global optimization problem, the raindrop updating operator \( \phi_U^F \) is executed as follows.

\[
\text{Pop}(t + 1) = \phi_U^F(\text{Pop}(t) \cup \text{Small Raindrop}(t)) \tag{13}
\]

That is to say; a sort method is used to select the \( N \) best individuals as the next population from the above two populations. In this paper, a bubble-sort procedure is applied for ranking method.

The flow chart of artificial raindrop algorithm is shown in figure 7.

From the figure 7 steps of ARD algorithm is noted and described as procedure as follows.

**6.1. Procedure of artificial raindrop algorithm**

Procedure of artificial raindrop algorithm is stated in 8 steps as follows,

1. **Initialization:** Select the algorithm parameters \( N \), \( \lambda \), \( D \), \( \tau_1 \), \( \tau_2 \), \text{Max Flow Number}; Randomly Generate initial population Pop(0); Set \( t = 0 \).

2. **Evaluation:** Estimate the objective function values of all vapors in Pop(0); Find the best solution gbest(0); RP(0) = gbest(0);
Step3. Raindrop Generation: Acquire Raindrop \( (t) \) by relating raindrop generation operator \( \phi_G^R \) to \( \text{Pop}(t) \);

Step4. Raindrop Descent: Acquire \( \text{New}_\text{Raindrop}(t) \) by relating raindrop descent operator \( \phi_D^R \) to \( \text{Raindrop}(t) \);

Step5. Raindrop Collision: Acquire \( \text{Small}_\text{Raindrop}(t) \) by relating raindrop collision operator \( \phi_C^R \) to \( \text{New}_\text{Raindrop}(t) \);

Step6. Raindrop Flowing: Acquire \( \text{New}_\text{Small}_\text{Raindrop}(t) \) by relating raindrop flowing operator \( \phi_F^R \) to \( \text{Small}_\text{Raindrop}(t) \);

Step7. Raindrop Updating: Acquire \( \text{Pop}(t+1) \) by applying raindrop updating operator \( \phi_U^R \) to \( \text{Small}_\text{Raindrop}(t) \) and update \( \text{Raindrop} \) pool \( \text{RP}(t+1) \);

Step8. Termination Test: If termination condition is satisfied, export the individual with the smallest objective function value in \( \text{Pop}(t+1) \), stop the algorithm; otherwise, \( t = t + 1 \), go to Step 3.

6.2. ARD algorithm in distillation

In a distillation column control, ARD algorithm is applied to tune the gains of PID controller in an outer loop of cascade control. \( K_p \), \( K_i \), and \( K_d \) are the output of ARD algorithm. \( K_p=0.312 \), \( K_i=0.069 \) and \( K_d=0.011 \) are the outputs ARD algorithm.

7. Simulation Results And Discussion

In a parallel cascade distillation set point of flow is set externally and is controlled by PID/FGS PID/ARD algorithm. Output of flow controller is act as a setpoint to the temperature controller. It is controlled using PID controller. For an analysis, flow is set in the range of 40 to 100 which is act as a reference value for temperature control. Figure 8 to 10 shows the performance flow using various controllers such as by PID, FGS PID and ARD algorithm.

Fig.8. Flow performance of PID-based parallel cascade control at 70mols/s

From the figure 8, it is noted that flow sets in its reference value of 70 after peak overshoot of 11.43%, which is very high and to be reduced.

Fig.9. Flow performance of FGS-based parallel cascade control at 70mols/s

From the figure 9, it is noted that peak overshoot is reduced compare to PID control. Meanwhile settling time is also reduced.

Fig.10. Flow performance of ARD algorithm based parallel cascade control at 70mols/s
From the figure 11 it is noted that effect of ARD controller in outer loop reduces overshoot in flow compare to inner loop temperature. From this it is noted that overshoot produced inner loop is also reduced by outer loop ARD controller.

From the figure 12, efficient performance of proposed system is noted that peak overshoot is effectively reduced compared to PID and FGS-PID controllers. Meantime steady state error is reduced compared to other controllers. The reduced peak overshoot and steady state error results in reduced ITAE compare to all other controllers as shown in Figure 13.

From the figure 13 it is noted that ITAE by ARD algorithm is very less due to effective control of flow. Comparative performance of all controllers in flow control at various flow levels are shown in figures 14 and 15.

From the figures 12,14 and 15 it is noted that at various flow levels ARD produces better performance than all other controllers. Comparative performance of PID based temperature control with various controllers in flow control is shown in figure16.
From the figure 16 it is noted that inner loop controller in all cases are PID controller, but the effect of outerloop controller results various level of control on a temperature as shown in figure 16. FGS PID controller in an outer loop reduces temperature overshoot compare to PID in an outerloop. ARD as flow controller reduces overshoot in a temperature compare to all other controllers. From this analysis it is noted that ARD produces better performance in flow as well as in temperature control.

Table 4 Trade-off of various controllers in distillation at flow 70mols/s

<table>
<thead>
<tr>
<th>Controller</th>
<th>Peak Overshoot (%)</th>
<th>Steady state error (%)</th>
<th>Rise time (sec)</th>
<th>Peak time (sec)</th>
<th>Settling time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID</td>
<td>11.48</td>
<td>0.417</td>
<td>26.8</td>
<td>61.8</td>
<td>147</td>
</tr>
<tr>
<td>FGS</td>
<td>8.57</td>
<td>0.34</td>
<td>28.8</td>
<td>61.2</td>
<td>139</td>
</tr>
<tr>
<td>ARD</td>
<td>4.54</td>
<td>0.28</td>
<td>38.8</td>
<td>71.2</td>
<td>139</td>
</tr>
</tbody>
</table>

Table 5 Trade-off of various controllers in distillation at flow 80mols/s

<table>
<thead>
<tr>
<th>Controller</th>
<th>Peak Overshoot (%)</th>
<th>Steady state error (%)</th>
<th>Rise time (sec)</th>
<th>Peak time (sec)</th>
<th>Settling time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID</td>
<td>7</td>
<td>0.31</td>
<td>26.8</td>
<td>61.8</td>
<td>147</td>
</tr>
<tr>
<td>FGS</td>
<td>3.5</td>
<td>0.24</td>
<td>28.8</td>
<td>61.2</td>
<td>139</td>
</tr>
<tr>
<td>ARD</td>
<td>1</td>
<td>0.17</td>
<td>38.8</td>
<td>71.2</td>
<td>139</td>
</tr>
</tbody>
</table>

Table 6 Trade-off of various controllers in distillation at flow 90mols/s

<table>
<thead>
<tr>
<th>Controller</th>
<th>Peak Overshoot (%)</th>
<th>Steady state error (%)</th>
<th>Rise time (sec)</th>
<th>Peak time (sec)</th>
<th>Settling time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID</td>
<td>5.7</td>
<td>0.3</td>
<td>26.8</td>
<td>61.8</td>
<td>147</td>
</tr>
<tr>
<td>FGS</td>
<td>3.3</td>
<td>0.21</td>
<td>28.8</td>
<td>61.2</td>
<td>139</td>
</tr>
<tr>
<td>ARD</td>
<td>0.93</td>
<td>0.12</td>
<td>38.8</td>
<td>71.2</td>
<td>139</td>
</tr>
</tbody>
</table>

From the table 4, it is noted that compare to PID controller around 60% overshoot and 33% of steady state error are reduced by ARD algorithm.

From the tables 4-6 it is noted that when flow increases, overshoot and steady state error are reduced. In all flow levels Peak overshoot and steady state are reduced by ARD algorithm compared to other controllers.

8. Conclusion

In this paper, a novel meta-heuristic method of artificial raindrop algorithm is proposed for control of flow with parallel cascade control of distillation column. Performance of proposed system is analyzed and compared with the performance of PID and FGS-PID based systems. FGS-PID controller reduces all parameters that were analysed when compare to PID based control. But from the analysis of FGS-PID, reduction in percentage of peak overshoot and steady state error is necessary. From the next set of analysis, it is noted that ARD algorithm control for flow produces better performance compared to PID and FGS-PID in aspects such as reduction in peak overshoot, peak time, rise time, settling time and steady state error. Peak overshoot and steady state error are reduced effectively by ARD algorithm which reduces the oscillation in flow and temperature of distillation column.

References