Design of Decentralized Fractional Order PI\(^{\lambda}\) Controller for Pilot Plant Binary Distillation Column

BETTOU KHALFA, CHAREF ABDELFATEH
dep. Electronique, Université des Frères Mentouri de Constantine,
Route Ain El-bey, 25240, Constantine, Algérie
bettou_kh@yahoo.com

Abstract: This paper presents the application of fractional order operators to improve the control quality of multivariable systems. The basic ideas of this tuning method are based, in the first place, on the existed tuning methods for setting the parameters of the decentralized fractional order PI\(^{\lambda}\) controller for \(\lambda=1\), which means setting the parameters of the classical decentralized PI controller, and the minimum integral criterion by using Particle Swarm Optimization (PSO) algorithm for setting the fractional integration action order \(\lambda\). The integral criterion is formulated to improve the dynamic response of the system, while causing a good decoupling between control loops. The Distillation Column, which is a multivariable system with two inputs and two outputs (TITO), in a decentralized control structure, is analyzed. Simulation results are presented to show the control quality improvement of this proposed decentralized fractional order PI\(^{\lambda}\) controller tuning method compared to the decentralized PI controller tuned using any existed tuning method.

Keywords: Decentralized PI\(^{\lambda}\) controller; Pilot plant binary distillation column; TITO; PSO; Integral criterion.

1. INTRODUCTION

Generally, most industrial processes are multivariable systems. Two-input two-output (TITO) system is one of the most prevalent categories of multivariable systems, because there are real processes of this nature or because a complex process has been decomposed in 2×2 blocks with non negligible interactions between its inputs and outputs [1, 2, 3]. When the interactions in different channels of the process are modest, a diagonal controller (decentralized control) is often adequate. Nevertheless, when interactions are significant, a full matrix controller (centralized control) is advisable.

Over the years, many studies have been developed in order to find new control strategies for providing a good dynamic behavior to multivariable systems. Researchers showed that, in many industrial plants, a strategy widely used is a decentralized approach of the conventional PI/PID algorithm, known as multi-loop PI/PID control [1].

The problem identified in this strategy is that this decomposition affects the dynamics of the multivariable process, since there is interaction between control loops [4]. One strategy that has been considered in the literature is the application of optimization heuristics to obtain the parameters for the controllers. This is due the ability of such methods in solving optimization problems (mostly nonlinear), with many constraints, resulting in optimized values [5].

In the last decades, there has been a great interest, both from an academic and industrial point of view, in fractional order controllers because, in principle, they provide more flexibility in the controller design [6, 7, 8, 9, 10]. The fractional order PI\(^{\lambda}\) controller is a PI controller with integral order is of fractional rather than integer. The extension of the integration order from integer to fractional order provides more flexibility in design of the controller, thereby controlling the wide range of dynamics of a system.

Different methods for the design of a fractional order PI\(^{\lambda}/\text{PI}^{\mu}\text{D}^{\nu}\) controller have been proposed in the literature, they are based on the minimization of different objective functions [11, 12, 13, 14]. In this paper, we propose the tuning of decentralized fractional order PI\(^{\lambda}\) controller. The proposed tuning strategy is based, in the first place, on any existed tuning methods for setting the parameters of decentralized PI controller. In this work, we have adopted an existed tuning method. Then using the decentralized PI controller obtained in the first step, an integral criterion is minimized to determine the optimum settings the fractional...
integration action order $\lambda$ of the decentralized fractional order PI$^\lambda$ controller. The PSO algorithm will be considered as search techniques for multi-loop fractional order PI$^\lambda$ controller tuning in one process, in order to minimize a cost function which enhances the dynamic response of the system, and causes a good decoupling between control loops.

2. TOOLS AND METHODS

Decentralized Fractional PI$^\lambda$ Control

First, conventional method of designing PI/PID type controllers for such Multi Inputs Multi Outputs (MIMO) processes require correct pairing of one manipulated variable to one controlled variable, to avoid poor controller performance and reduced stability margins, which can be achieved by means of Relative Gain Array (RGA) approach. Other improved measures of loop interaction, necessary and sufficient conditions for pairing, control structure selection etc. have been thoroughly reviewed in [15], though the RGA based loop interaction analysis still dominates the process control industries. Only steady-state information about the process gain matrix is required to develop a RGA, which provides a measure of process interactions between the manipulated variables and controlled variables. The most effective pairing can be achieved if a manipulated variable is used to monitor a controlled variable with which its measure of interaction is highest, preferably, close to unity. This allows pairing of a single controlled variable with a single manipulated variable via a feedback controller for two such loops of TITO process as in Fig. 1 with conventional PI controller.

The proposed decentralized fractional order PI$^\lambda$ controller for a TITO process has two fractional order PI$^\lambda$ controllers. The decentralized fractional order PI$^\lambda$ controller combines these two individual fractional order PI$^\lambda$ controllers together in an appropriate way. Therefore, the control structure shown in Fig. 1, which is known as a decentralized fractional order PI$^\lambda$ control structure, is proposed.

![Fig. 1 The decentralized fractional order PI$^\lambda$ control structure.](image)

The most common form of a fractional order controller is the PI$^\lambda$ controller, involving an integrator of order $\lambda$ where $\lambda$ can be any real numbers. The transfer function of such a controller has the following form:

$$C(s) = \left( K_p + K_i \frac{1}{s^\lambda} \right)$$  \hspace{1cm} (1)

With $K_p$, $K_i$ are proportional and integral gains, respectively.

Clearly, selecting $\lambda = 1$, a conventional PI controller can be recovered. One of the most important advantages of the fractional order PI$^\lambda$ controller is the possible better control of fractional order dynamical systems. Another advantages lies in the fact that the fractional order PI$^\lambda$ controllers are less sensitive to changes of parameters of a controlled system. This is due to the extra degree of freedom to better adjust the dynamical properties of a fractional order control system.

Rational function of the fractional PI$^\lambda$ controller

When fractional order controllers have to be implemented or simulations have to be performed, fractional order transfer functions are usually replaced by integer transfer functions whose behavior is close enough to the desired ones but much easier to handle. There are many different ways to get such approximations but unfortunately it is not possible to say that one of them is the best, because even though some of them are better than others in regard to certain characteristics, the relative merits of each approximation depend on the differentiation order, on whether one is more interested in an accurate frequency behavior or in accurate time responses, on how large admissible transfer functions may be, and
In Particle Swarm optimization, the neural network training, the rational function approximation of the fractional order integrator can be calculated as given in [16], [17].

Partial swarm optimisation

Optimization, today, is a concept widely discussed in scientific research communities, because its application is given to obtain the best performances in several problems. Optimization aims to improve the performance of a system in a specific situation, modeled as a cost function with its constraints.

Within this context the heuristics methods are inserted. They represent a good estimate for searching optimal solutions. The concept of heuristics is to seek for solutions using an intuitive analysis, ensuring a substantial reduction in the complexity of searching problems, without a deep knowledge.

There are several heuristics developed in the literature, but for this paper the focus will be given to PSO.

Particle Swarm Optimization algorithm is an intelligent optimization algorithm intimating the bird swarm behavior which was proposed by psychologist Kennedy and Dr. Eberhart in 1995 [18]. Compared to other optimization algorithms, the Particle Swarm Optimization is more objective, easy and performs well. It is applied in many fields such as the function optimization, the neural network training, the fuzzy system control, etc. In Particle Swarm Optimization algorithm, each individual is called “particle” which represents a potential solution. The algorithm achieves the best solution by the variability of some particles in the tracing space. The particles search in the solution space following the best particle by changing their positions and the fitness frequently; the flying direction and velocity are determined by the objective function.

Assuming \( X_i = (x_{i1}, x_{i2}, ..., x_{id}) \) is the position of the particle in D-dimension, \( V_i = (v_{i1}, v_{i2}, ..., v_{id}) \) is its velocity which represents its direction of searching. In iteration process, each particle keeps the best position \( p_{best} \) found by itself, besides, it also knows the best position \( g_{best} \) searched by the group particles, and changes its velocity according two best positions. The standard formula of Particle Swarm Optimization is as follow:

\[
\begin{align*}
X_{id}^{k+1} &= X_{id}^k + V_{id}^{k+1} \\
V_{id}^{k+1} &= wV_{id}^k + c_1r_1(p_{id} - X_{id}^k) + c_2r_2(g_{id} - X_{id}^k)
\end{align*}
\]

where: \( i=1,2,...,N \); \( N \) the number of the group particles; \( d=1,2,...,D; \) \( k \) the maximum number of iteration; \( r_1 \) and \( r_2 \) the random values in \([0,1]\) used to keep the diversity of the group particles; \( c_1 \) and \( c_2 \) the learning coefficients, also they are called acceleration coefficients; \( V_{id}^k \) the number \( d \) component of the velocity of particle \( i \) in \( k \)-th iteration; \( X_{id}^k \) the number \( d \) component of the position of particle \( i \) in \( k \)-th iteration; \( p_{id} \) the number \( d \) component of the best position particle \( i \) has ever found; \( p_{gd} \) the number \( d \) component of the best position the group particles have ever found; \( w \) denotes the inertia weight factor.

In order to prevent velocity extrapolation, the maximum speed \( V_{max} \) and minimum speed \( V_{min} \) are added to the model, such that:

\[
\begin{align*}
\text{If } v_{id}^{k+1} \geq v_{max}, & \quad \text{then } v_{id}^{k+1} = v_{max} \\
\text{If } v_{id}^{k+1} \leq v_{min}, & \quad \text{then } v_{id}^{k+1} = v_{min}
\end{align*}
\]

A restriction in the particles’ position is also imposed, \( x_{max} \) and \( x_{min} \), in order to prevent the distant solution.

\[
\begin{align*}
\text{If } x_{id}^{k+1} \geq x_{max}, & \quad \text{then } x_{id}^{k+1} = x_{max} \\
\text{If } x_{id}^{k+1} \leq x_{min}, & \quad \text{then } x_{id}^{k+1} = x_{min}
\end{align*}
\]
The procedure of standard Particle Swarm Optimization is given as following:
Step 1: Initialize the original position and velocity of particle swarm;
Step 2: Calculate the fitness value of each particle;
Step 3: For each particle, compare the fitness value with the fitness value of pbest, if current value is better, then renew the position with current position, and update the fitness value simultaneously;
Step 4: Determine the best particle of group with the best fitness value, if the fitness value is better than the fitness value of gbest, then update the gbest and its fitness value with the position;
Step 5: Check the finalizing criterion, if it is satisfied, quit the iteration; otherwise, return to step 2.

3. TUNING OF DECENTRALIZED FRACTIONAL PI CONTROLLER

Tuning of parameters $K_p$ and $K_i$

Our tuning method is based, in The first place, on any tuning method for setting the parameters $K_p$ and $K_i$ of the decentralized fractional order PI controller for $\lambda=1$ which means setting the parameters of a simple decentralized PI controller.

Tuning of the parameter $\lambda$

With the parameters $K_p$ and $K_i$ obtained in the first step, we use PSO algorithm to determine the optimum setting of the fractional integration action order $\lambda$ of the decentralized fractional PI controller. The integral criterion to be minimized is defined as:

$$J(\lambda_1, \lambda_2) = \int e_1(t) dt + \int e_2(t) dt + \int y_1(t) dt + \int y_2(t) dt$$

With $e_{1,2} = [r_{1,2}(t) - y_{1,2}(t)]$ is the error signals, $y_{1,2}$ and $y_{21}$ are outputs of interaction loops.

4. SIMULATION EXAMPLE

This section shows the application of the results obtained for the tuning parameters of the decentralized fractional order PI$^h$ controller using a selected TITO process. The distillation column process is a multivariable system that has been studied extensively.

Distillation is meant for the separation of two different mixtures based on its boiling points of the solution. It is the most widely used separation process in petroleum industries. The choice of the best configuration for dynamic control of distillation columns is a major concern, and considerable research activity has been devoted to finding the best control configuration [19].

The current article considers the transfer function model for simulation. The reflux flow rate and reboiler power rate are the Manipulated Variables (MV), whereas the tray temperature is considered as the Control Variable (Temp. of T5 and Temp. of T1) for the present simulation work [20].

The process has the following transfer matrix:

$$G(s) = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}$$

$$g_{11}(s) = \frac{-0.16e^{-0.01s}}{0.01s+1}$$

$$g_{12}(s) = \frac{0.6e^{-1.9s}}{0.05s+1}$$

$$g_{21}(s) = \frac{-0.04e^{-0.01s}}{0.02s+1}$$

$$g_{22}(s) = \frac{0.49e^{-0.47s}}{0.19s+1}$$

The steady state gain matrix of the TITO system is given by

$$G(s = 0) = \begin{bmatrix} -0.16 & 0.6 \\ -0.04 & 0.49 \end{bmatrix}$$

As previously mentioned, proper pairing of manipulated variable with the controlled variable is required to minimize the effect of loop interaction as much as possible for the design of decentralized controllers for multivariable processes.

The common criterion, used to obtain the knowledge of correct pairing is the relative gain array or RGA, derived from the dc gain $G(0)$ of the steady state process transfer matrix as:

$$RGA = G(0) \otimes (G(0)^{-1})^T$$

Where, $\otimes$ denotes the element-by-element multiplication of the matrices. The RGA for the process is:
It is observed from the RGA values of the TITO process that the process suffers from high loop interactions. Hence designing decentralized controllers by pairing of any manipulated variable with any controlled variable will not lead to a satisfactory performance. In this paper, we utilized simultaneous tuning of both the fractional order PI\(^\delta\) controllers at a time for considering the effect of loop interactions in the tuning phase while minimizing the objective function (6) for all loops, instead of tuning one controller as in a SISO loop. The above discussed PSO algorithm has now been applied to tune decentralized fractional order PI\(^\delta\) controller parameters for the TITO process (7). In our simulations, the PSO algorithm is repeated 20 times, the minimum criteria give the optimum controller parameters. Table 1 presents the PSO parameters used in the simulation.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of iterations</td>
<td>100</td>
</tr>
<tr>
<td>Number of trials</td>
<td>20</td>
</tr>
<tr>
<td>Swarm size (N)</td>
<td>25</td>
</tr>
<tr>
<td>Learning coefficients (c1=c2)</td>
<td>2.49618</td>
</tr>
<tr>
<td>Inertia weight factor (w)</td>
<td>0.7298</td>
</tr>
</tbody>
</table>

The performance of firefly algorithm in term of fitness function and iterations is shown in Fig.2.

The unknown parameters of the fractional PI\(^\delta\) controllers are recorded at the end of iteration 100, where, the process of optimization gets terminated. Variations of the decentralized fractional order PI\(^\delta\) controller in terms of iterations for \(\lambda_1\) and \(\lambda_2\) are shown in Fig. 3 and Fig.4, respectively. It is clear that PSO algorithm is able to set all fractional orders of the decentralized fractional order PI\(^\delta\) controller in almost 100 iterations.

The set-point tracking performances of the first loop have been shown in Fig. 5 and Fig.6, respectively. The set-point tracking performances of the second loop have been shown in Fig. 7 and Fig.8, respectively. The performance measure \(J(\lambda)\) is calculated for the considered operating conditions and the results are tabulated in table 2. The decentralized fractional order PI\(^\delta\) controller has better performance over the conventional decentralized PI controller. Tables 3 and 4 report the best found optimum controller parameters for the two loops of the TITO process using PSO algorithm.
Table 2  Performance measure J for TITO control

<table>
<thead>
<tr>
<th>Control structure</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decentralized PI control</td>
<td>13.6975</td>
</tr>
<tr>
<td>Decentralized fractional PI control</td>
<td>4.9087</td>
</tr>
</tbody>
</table>

Table 3  Tuned controller for the first loop

<table>
<thead>
<tr>
<th>Controller</th>
<th>PI[20]</th>
<th>FPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_{P_1}$</td>
<td>-0.1</td>
<td>-0.1</td>
</tr>
<tr>
<td>$K_{I_1}$</td>
<td>-6.67</td>
<td>-6.67</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>1</td>
<td>1.0958</td>
</tr>
</tbody>
</table>

Table 4  Tuned controller for the second loop

<table>
<thead>
<tr>
<th>Controller</th>
<th>PI[20]</th>
<th>FPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_{P_2}$</td>
<td>0.4</td>
<td>0.4</td>
</tr>
<tr>
<td>$K_{I_2}$</td>
<td>0.53</td>
<td>0.53</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>1</td>
<td>1.1062</td>
</tr>
</tbody>
</table>

It is to be noted that some of the obtained controller gains are negative which is due to the fact that the TITO process has negative transfer functions in some loops.

For control performances enhancement comparison, we have summarized some temporal performances characteristics in table 5 and 6 for the feedback control TITO system with both decentralized controllers.

In short, we can say that the proposed decentralized fractional order PI* controller outperforms the conventional decentralized PI controller, and an acceptable step response is traced with good decoupling between loops.

Table 5  Temporal characteristics of the first loop

<table>
<thead>
<tr>
<th>Controller</th>
<th>Settling time (sec)</th>
<th>Overshoot %</th>
</tr>
</thead>
<tbody>
<tr>
<td>PI[20]</td>
<td>10.6</td>
<td>0.00</td>
</tr>
<tr>
<td>FPI$_1$</td>
<td>3.28</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 6  Temporal characteristics of the second loop

<table>
<thead>
<tr>
<th>Controller</th>
<th>Settling time (sec)</th>
<th>Overshoot %</th>
</tr>
</thead>
<tbody>
<tr>
<td>PI[20]</td>
<td>24.00</td>
<td>0.00</td>
</tr>
<tr>
<td>FPI$_2$</td>
<td>2.93</td>
<td>0.92</td>
</tr>
</tbody>
</table>

Fig. 5  Step responses of the first closed loop systems with $C(s)$ as a conventional PI controller (dashed line) and $C(s)$ as a fractional PI* controller (solid line).

Fig. 6  Step responses of the first interaction loop systems with $C(s)$ as a conventional PI controller (dashed line) and $C(s)$ as a fractional PI* controller (solid line).

Fig. 7  Step responses of the second closed loop systems with $C(s)$ as a conventional PI controller (dashed line) and $C(s)$ as a fractional PI* controller (solid line).

Fig. 8  Step responses of the second interaction loop systems with $C(s)$ as a conventional PI controller (dashed line) and $C(s)$ as a fractional PI* controller (solid line).

5. CONCLUSION

In this paper, a new strategy was proposed for the decentralized fractional order PI* controller tuning for MIMO process based on existed decentralized PI controller. The optimization based controller parameter
selection uses an objective function as integral criterion which enhances the
dynamic response of the system, and causes a
good decoupling between control loops.
The minimization problem was solved using
the popular Particle Swarm Optimization
algorithm. The formulations of this new
conception strategy have been derived using
the rational function approximation of the
fractional integrator and differentiator
operators, in a given frequency band of
practical interest.
An illustrative example was presented to
show the effectiveness and the simplicity of
the proposed method. The obtained results,
in terms of time performances, using the
optimal decentralized fractional order PI
controller have compared to the ones of the
conventional decentralized PI in distillation
column plant. By using the decentralized
fractional order PI controller, we have
significantly reduced the integral criterion J,
and the settling time of the feedback control
system.
Our design method will be very suitable for
already conventional tuned PI controller for
multivariable systems. Implementation
considerations of the fractional PI
controller with already existing PI controller for
multivariable systems will be presented in future
works.
Our further research efforts will include
testing on more type’s criterions and testing
the proposed control structure in an
experimental platform.

References


