EVALUATION AND OPTIMIZATION OF THREE-PHASE SEPARATOR VESSEL CONTROLLED BY PI CONTROLS APPLYING PARTICLE SWARM OPTMIZATION

André Felipe de Azevedo Dantas [*]
Leandro Luttiane da Silva Linhares [*]
Jan Erik Mont Gomery Pinto [*]
Fábio Meneghetti Ugulino de Araújo [*] **André Laurindo Maitelli** [*]

[*] *Automation Laboratory in Oil*
Federal University of Rio Grande do Norte, Natal/RN - Brazil

Abstract: The oil platforms are large structures with a large amount of control loops based on PID controllers. Over time, even when they are tuned correctly at the beginning of its operation, their performance deteriorates due to changes in the dynamics of the processes, making it important to evaluate the control loop and retune PID controllers when necessary. In this work we propose a combination of the CPSO and GPSO algorithms to tune PID controllers in a primary separation system of a primary platform. The tuning and optimization are performed through the control loop evaluation indices IAE, ISE, ITAE, ITSE and Goodhart.

Keywords: PI controller, Global optmization, Performance indices, Level control

1. INTRODUCTION

The effective separation of produced fluids in the extraction of oil has long been a major challenge in the oil industry (Behin and Aghajari, 2008). Once removed from the wells, extraction of the produced fluids are transferred to a separation system (Silva *et al.*, 2000).

Even if the separation between the components contained in the oil occurs at stations or in the production unit is necessary to process and refine the mixture from the reservoir rock, in order to get their products (Silva *et al.*, 2000). This procedure is done in gravitational three-phase separator vessel, target of study.

Therefore it is necessary to evaluate the mesh in order to understand if there is need for improvement in control processes. This paper proposes the evaluation of control loops using the classical methods ITE, ITSE, ITAE, IAE and Goodhart, associated with an optimizer whose objective is to minimize the values measured by the indices. The optimizer is a junction of algorithms GPSO and CPSO.

2. SEPARATOR VESSEL PROCESS

The separator vessel are equipment of large dimensions that performs a separation of the phase aqueous, oil phase and gas phase, keep them within of tolerance limits, they can be biphasic and three-phase. To do this requires a long period of time so that by the gravitational action, it is possible that separation. For the case of three-phase separator which performs the separation of the phases, it is necessary to maintain the limits of the quantity of liquid together in the gas, the quantity of water together in the oil and oil together in the water within tolerable regions ensuring the quality process (Silveira, 2006).

An important element that difficult the control in the separator vessel is the gushes. They are generated when the fluid arrives in the separator vessel, which is formed of dispersed phases of water, oil and gas. The evolution of the fluid flow passes by flow instability theta can occur at certain flow, due an arrangement of the set line-riser adverse, generally at low flow rates in relatively long lines (Silveira, 2006). Besides being used separator vessels can also be used in conjunction hydrocyclones.

The hydrocyclone is a process of oil/water separation most commonly used by the oil industry today, because it´s compact and efficient in meeting environmental requirements for ejectment of water for the environment (Filgueiras, 2005). The equipment includes cylindrical and conical portions juxtaposed, having tangential inlets for the case where the mixed fluid inlet and a bottom outlet and an upper outlet, where each one is used by a different fluid (Júnior, 2008).

Within this context we developed a process simulator designed in order to represent phenomena that occur in a three-phase separator vessel, in hydrocyclones and gas-lift respectively. The system was implemented on file extension *.m* of the Matlab®, interacting, after being drawn with the Simulink® program used for simulations.

It´s possible view in the **Figura 14** wells with continuous gas-lift that feeding the three-phase separator with water, oil and gas, PI controllers, used to control the water level, oil level, pressure in the separator vessel, pressure and pressure difference in the hydrocyclone.

3. PARTICLE SWARM OPTIMIZATION

The Particle Swarm Optimization (PSO) algorithmm, originally introduced by (Kennedy and Eberhart, 1995), has become one of the most important swarm intelligence-based algorithms. Due to its simple implementation, minimum mathematical processing and good optimization capability, PSO has attracted more attentions. As (Vassiliadis and Dounias, 2009) summarized, PSO can be considered the most popularly and frequently applied among nature-inspired techniques according to related publications.

PSO simulates the behavior of a bird flock. When a flock of birds forage for food, two simple and important strategies are: (a) searching for the peripheral region around the bird that is nearest to the food; (b) judging the position of the food by its own flying experience. Inspired by the two strategies, the search space of optimization problem is regarded as birds´flying space; every bird is abstracted as a particle with non-quality and non-volume so as to denote a candidate solution; and the optimal solution to be searched for the problem is the food to all the particles. Then, the basic PSO algorithm comprises a swarm of particles moving in the D-dimensional search space which includes all possible candidate solutions (Shi *et al.*, 2011).

We denote the ith particle as $X_i = (x_{i,1}, x_{i,2}, \cdots, x_{i,d})$, and its flying velocity as $V_i = (v_{i,1}, v_{i,2}, \cdots, v_{i,d})$. Depending on two strategies above, when each particle flies in the search space, $P_i = (p_{i,1}, p_{i,2}, \cdots, p_{i,d})$ denotes the personal best position the ith particle has found so far, and $P_g = (p_{g,1}, p_{g,2}, \cdots, pg, d)$ is the global best position discovered by the swarm. At each time step t, both of each particle's velocity and position are updated so that a particle moves to a new position. The following two equations are employed to calculate the velocity and position:

$$V_i^{t+1} = V_i^t + c_1 r_1 \left(P_i^t\right) - X_i^t + c_2 r_2 \left(P_g^t - X_i^t\right) \tag{1}$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \tag{2}$$

where c_1 and c_2 are two positive constants (acceleration constants), r_1 and r_2 are two uniform random numbers in $[0, 1]$. A constant V_{max} is often used to limit each particle's velocity to guarantee that most of the new positions of the particles will be restricted in the search space of feasible region at each iteration. Equations 1 and 2 lead to the movement of each particle to wards its cognitive best position (P_i) and "social" best position (P_g) with random perturbations caused by $c_1 r_1$ and $c_2 r_2$, respectively. The quality of a particle's position is measured by its fitness value, namely, the better fitness value means the better position.

(Shi and Eberhart, 1998) modified the original PSO by introducing an inertia weight (ω) to Equation 1 to balance exploitation and exploration. The modified Equation 1 is as follows:

$$V_i^{t+1} = \omega V_i^t + c_1 r_1 \left(P_i^t\right) - X_i^t + c_2 r_2 \left(P_g^t - X_i^t\right) \tag{3}$$

According to the numerical tests in (Shi and Eberhart, 1998; Shi and Eberhart, 1999), the adoption of ω improves the performance of PSO largely, so the combination of Equations 3 and 2 is always considered as the standard PSO by many researchers.

The advantage of PSO algorithm is quick in the convergence speed, few in the parameters, simple in operations and etc., therefore it is suitable to solve optimization problems. The PSO algorithm, however, is easy to appear premature convergence, especially in multimodal cases. To overcome these defects, many scholars proposed the improvement methods on PSO algorithmThose improvement methods can be sorted to two kinds. The first kind is to combine PSO with other optimization algorithms. The second kind is to improve the structure of PSO algorithm (speed iteration formula and position iteration formula), specially speed iteration formula (Baiquan *et al.*, 2012).

In this paper we combine the CPSO–*outer* proposed by (Shi *et al.*, 2011) with the improved speed formula of the GPSO proposed by (**?**). The cellular particle swarm optimization (CPSO) is a hybridization of cellular automata (CA) and particle swarm optimization (PSO) for function optimization.

In order to enhance the performance of the interaction, the CPSO employs the idea of CA to explore the communication structure and information inheriting and diffusing mechanisms of the swarm system of PSO. So in CPSO, we consider PSO with a CA model, where individuals can only exchange information within their neighborhood. It could help exploring the search space due to the slow information diffusion through the population, promoting the preservation of diversity, and exploiting every cell's local information inside the neighborhood (Shi *et al.*, 2011).

The ith particle's position X_i^t is defined as the cell state of this particle. When implementing CPSO-outer, only smart-cells participate in updating process. A smart-cell constructs its neighborhood by a neighborhood function given in Equation 4.

$$N(i) = \begin{cases} X_i^t + A(i)R_3 \circ V_i^t \\ \quad iffit(X_i^t \neq, fit(P_g^t) \geq 0 \\ X_i^t + B(i)R_3 \circ V_i^t \\ \quad iffit(X_i^t \neq, fit(P_g^t) < 0 \\ X_i^t + C(i)R_3 \circ V_i^t \\ \quad iffit(X_i^t =, fit(P_g^t) \geq 0 \\ X_i^t + C(i)R_3 \circ V_i^t \\ \quad iffit(X_i^t =, fit(P_g^t) < 0 \end{cases} \tag{4}$$

$$\begin{aligned} A(i) &= \frac{fit(P_g^t}{fit(X_i^t)} \\ B(i) &= |\frac{fit(X_i^t}{fit(P_g^t)}| \\ C(i) &= (\frac{\exp fit(P_g^t)}{\exp fit(X_i^t)}^2) \end{aligned} \tag{5}$$

where R_3 is a $1 \times d$ matrix composed by d uniform random numbers in $[-1, 1]$, and $\circ$ is the operation symbol of Hadamard product. The transition rule of CA in CPSO-outer is designed as follows:

$$\begin{aligned} f(\phi) = min(&fit(N(i), \\ &fit(N(i+\delta_i)), \\ &\cdots, fit(N(i+\delta_m), \\ &\cdots, fit(N(i+\delta_l))) \end{aligned} \tag{6}$$

where

$$\phi = \begin{cases} i & x \leq iff(\phi) = fit(N(i)) \\ i + \delta_m & f(\phi) = fit(N(i+\delta_m)) \end{cases} \tag{7}$$

$$S_i^{t+1} = S_\phi^t \tag{8}$$

Equation 6 means that l neighbors of the ith particle generated by Equation 8 are evaluated, and the neighbor with the best fitness value is chosen to replace the ith particle. The transition rule could give particles the power to make a wise jump, and it could help exploring the search space in a local competition neighborhood and enhance the diversity of the swarm. So CPSO-outer could have greater potential to search for global optimum in the search space (Shi *et al.*, 2011).

In the algorithm applied in this work the standard position update formula was used. To each particle, neighbors were generated using the CPSO-concept. The particle fitness is compared with the its fitness' neighbors an replaced if a better neighbor is found.

To update the particle velocity, the improved velocity formula proposed by GPSO was used. It is described as in Equations 9, 10 and 11.

$$\begin{aligned} V_i^{t+1} = \omega \check{V}_i^t &+ c_1 r_1 (P_i - X_i^t) \\ &+ c_2 r_2 (P_g - X_i^t) \end{aligned} \tag{9}$$

$$\check{V}_i^{t+1} = k_0 V_i^{t+1} + r_3(\check{V}_i^t - V_i^{t+1}) \tag{10}$$

$$X_i^{t+1} = X_i^t + \check{V}_i^{t+1} \tag{11}$$

where adjustment coefficient k_0 is a constant in the range $[0, 1]$, r_3 is a random number with uniform distribution on the range $[0, 1]$, the other parameters are the same with the standard PSO.

4. RESULTS

The analysis developed in this study observed aspects of evaluation and tuning of meshes from the algorithm that uses optimization by particle swarm.

In principle, the algorithms were developed to evaluate the meshes (ITE, ITA, ISE, ITSE and Goodhart). These algorithms have been entered in the model with three-phase separators so that it was possible to evaluate each individual mesh, the total of 6 mesh evaluated, and all together represented by the sum of the evaluations. After, PI controllers were designed following the methodology of Teixeira and reproduced the same results of the tuning PI controllers. The tuning can be view in the Table 4.

Table 1. Original PI controllers parameters

PI	textbfTuning	**Original**
	P	I
Pi (S_u)	366.228	-0.51559
Pi 2(S_g)	366.228	-2.77494
Pi 3(S_{o3})	16.6667	-0.096
Pi 4(S_{o2})	8	0.05
Pi 5(S_{o1})	366.228	-2.77494
Pi $n^o 6(S_l)$	366.228	-2.77494

Then, the control loops were evaluated in a global way, with the methods listed in order to obtain the following results in Table 4.

Table 2. Control loops evaluation for original PI parameters

Method	**Evaluation**
iae	0,0545
ise	0,0099
itae	769,2202
itse	135,2689
goodhart	0,0523

From the displayed tuning with the respective values, was implemented a tuning using the PSO algorithms displayed in the PSO section. This algorithm considered that the first boot of the particles was at one point around the tuning shown and that from it there is that the optimization. To perform the optimization was also assumed that the system would be acting regulatory points mode of operation shown in Table 4.

Table 3. Setpoint values of the controlled variables

Output	Setpoint
Oil level	$0,5$
Water level	$0,5$
Vessel pressure	$9,4806$
Hydrocyclones differential pressure	$1,25$

After optimization, used as a function of minimization the methods for evaluation of meshes already listed in order to obtain the results in Table 4.

Table 4. Control loops evaluation for optmized PI parameters

Method	Evaluation
iae	0,0409
ise	0,0075
itae	528,5356
itse	92,8278
goodhart	0,0500

Concluding the results, a figure showing the difference between the graphics optimized system and the not optimized system from the perspective of output and control signal can be viewed in Figures 2 and **??**.

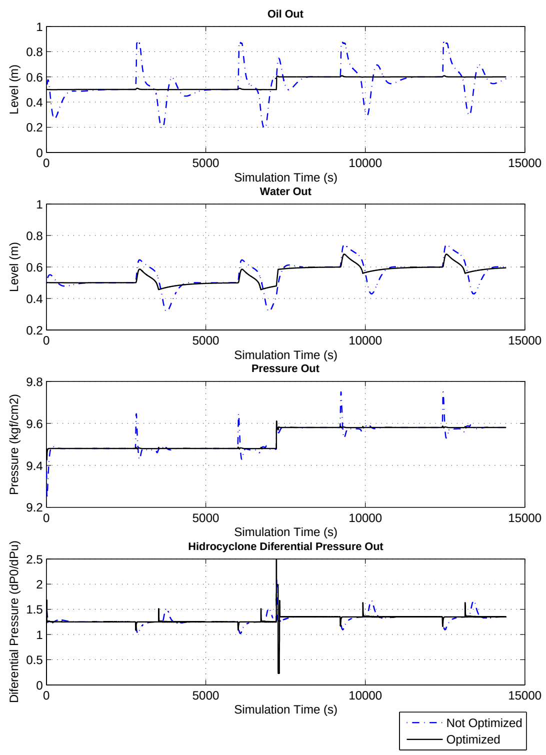

Fig. 1. Comparing the outputs of the optimized controls and the original tuning

In the Figures 2 and **??** displayed, simulation was performed for the system regulation, as well as changes were made in the references around the operating

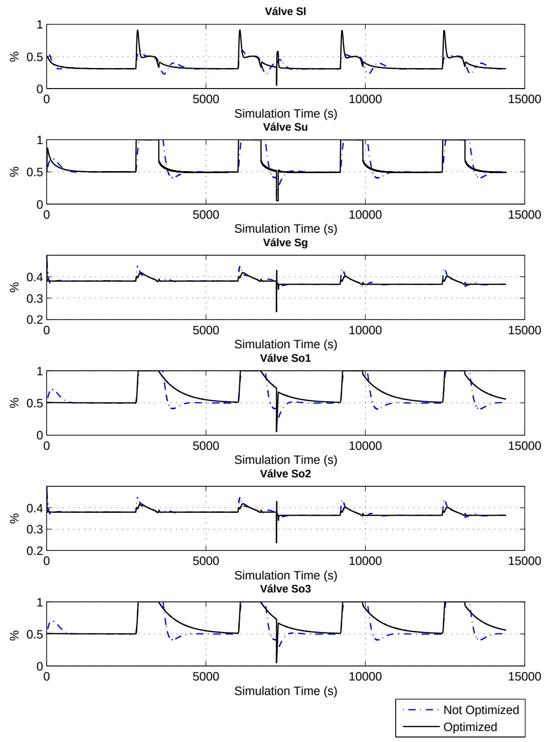

Fig. 2. Comparing the signal controls of the optimized controls and the original tuning

point, extrapolating in the practical the limits of retuning.

The Figures 2 and **??** show that optimization performed is compatible with the reduction of the indexes shown, which meant an improvement in the regulatory system and besides the improvement of the regulatory system was also perceptible that for a small variation in the reference, the system still tends to have consistent performance for optimization performed.

5. CONCLUSIONS

A method of controllers PID's retuning was shown, using like loss function standard methods of evaluation of meshes in the literature. For the three-phase separation system were compared with the simulated tuning presented in a literature and the optimized with PSO. It was possible to see that using as input to of the evaluation methods mentioned, there was a significant improvement of the tuning shown, representing a regulatory less sensitive to external disturbances as well as allowing small changes in setpoints, although it has been tuned to optimally mode regulatory.

6. ACKNOWLEGMENTS

The authors would like to thank Petrobras, Human Resources Program (HRP), NUPEG PRH-14 ANP, National Petroleum Agency and Coordination for the

Improvement of Higher Level or Education Personnel (CAPES) for the financial support.

7. REFERENCES

Baiquan, L., G. Gaiqin and L. Zeyu (2012). The block diagram method for designing the particle swarm optimization algorithm. *Journal of Global Optimization* **52**(4), 689–710.

Behin, J. and M. Aghajari (2008). Influence of water level on oil-water separation by residence time distribution curves investigations, separation and purification technology. *Separation and purification technology* **64**, 48–55.

Filgueiras, N.G.T. (2005). Modelagem, análise e controle de um processo de separação óleo/água. Master's thesis. UFRJ. Rio de Janeiro - Brazil.

Júnior, C. A. Corrêa (2008). Desenvolvimento de modelo computacional de previsão de quebra de gotas em simulador de separação de óleo e água em um hidrociclone. Master's thesis. UERJ. Rio de Janeiro.

Kennedy, J. and R.C. Eberhart (1995). Particle swarm optimization. In: *Proceedings of IEEE International Conference on Neural Networks*. pp. 1942–1948.

Shi, Y. and R.C. Eberhart (1998). A modified particle swarm optimizer. In: *Proceedings of IEEE International Conference on Evolutionary Computation*. pp. 66–73. Anchorage, Alaska.

Shi, Y. and R.C. Eberhart (1999). Empirical study of particle swarm optimization. In: *Proceedings of IEEE International Conference on Evolutionary Computation*. pp. 1945–1950.

Shi, Y., H. Liu, L. Gao and G. Zhang (2011). Cellular particle swarm optimization. *Information Sciences, Special Issue on Interpretable Fuzzy Systems* **181**(20), 4460–4493.

Silva, C.B.C., M.J.B. Filho and J.A. Pinheiro (2000). Medição de vazão e propriedades em escoamento multifásico: Solução econômica para diferentes atividades industriais. In: *Boletim Técnico PETROBRAS*. Vol. 43. PetrobrÁs. Rio de Janeiro. pp. 45–61.

Silveira, M.A.C.R. (2006). Controle de um processo de tratamento primário de petróleo. Master's thesis. UFRJ. Rio de Janeiro - Brazil.

Vassiliadis, V. and G. Dounias (2009). Nature-inspired intelligence: a review of selected methods and applications. *International Journal on Artificial Intelligence Tools* **18**(4), 487–516.