Journal of Petroleum Exploration and Production Technology (2020) 10:501–514
https://doi.org/10.1007/s13202-019-00746-x

ORIGINAL PAPER - EXPLORATION ENGINEERING

Construction of software using gain-schedule/auto-adaptive control strategy for automation in drilling fluid production plants

S. C. Magalhães[1] · G. Fontella[1] · R. F. O. Borges[1] · M. P. Vega[1] · L. A. Calçada[1] · C. M. Scheid[1]

Received: 21 March 2018 / Accepted: 24 July 2019 / Published online: 9 August 2019
© The Author(s) 2019

Abstract

Although many advanced nonlinear process control techniques have been developed over the past decade, classic control based on feedback response still has its place. This is mostly so because feedback empirical control is robust and simple to implement and does not require fancy calculations or high-qualified operational manpower to operate it. This work has developed an application written in LabVIEW® environment capable of doing a fully automated single-input single-output control. Preliminary tests were performed in a drilling fluid production unit, controlling flow rate through manipulation of the pump power engine. In the future, tests in the same plant of pressure control by choke valves manipulation will be performed (as found in rig sites, where wellbore pressure is controlled by manipulation of such valves). The final goal is to implement such software in a real rig site, to help operators in drilling control areas such as flow rate and wellbore pressure. The produced software has embedded three self-developed features: automatic plant identification (API), auto-tuning (ABAP) and controllers auto-switch (CAS). The API determines automatically the linearity of the process determining the empirical parameters according to Sundaresan and Krishnaswamy technique. In sequence, it calculates the parameters for P, PI and PID controllers using Cohen–Coon and Ziegler–Nichols methods. The API method automates the sequence of tests necessary to implement the Sundaresan and Krishnaswamy empirical approach. The ABAP feature based on heuristic rules tunes in real time the controllers' parameters to optimize its response. The CAS allows automatic switch between controllers and parameters to avoid instability, overshoots and creates a synergy with ABAP feature. The results have shown that the API feature is a good optimizer reducing the invested time to calculate all the parameters, from hours to a few minutes. The CAS results demonstrated an associative property with the ABAP feature to mitigate instabilities and overshoots. Therefore, the preliminary results suggested this software is a unique and important tool to improve performance, profitability and reliability during offshore and onshore drilling operations. Moreover, this application could be used in any industry with an approximate first-order dynamic system due to its robustness and a low human interaction need.

Keywords Automatic control · Auto-tuning · Feedback adaptive control · PID · Heuristic rules · Drilling fluid control

Introduction

During the exploration process of oil wells, drilling operations are a critical stage and, due to its complexity, require high investments. This step involves the weight application on a rotating drill string, which causes the destruction of the geological formation linking the surface to the oil reservoir. The use of a drilling fluid is necessary mostly to chill the drill system, maintain pressure between wellbore and rock formation and remove drill cuttings from the bottom of the well, among others functions (Bourgoyne et al. 1991).

Drilling fluids are commonly heavy, opaque and abrasive. Besides that, drilling fluids have complex rheological behavior and tend to cause most online sensors to fail due to the high concentration of solids suspended (Caenn and Chillingar 1996; Magalhães et al. 2014). Therefore, monitoring and controlling operational conditions on such fluid are not trivial (Broussard et al. 2010). If a failure in hydraulic pressure control occurs, serious operational problems such as loss circulation, inefficient well cleaning or even a blowout may arise (Gandelman et al. 2013).

✉ S. C. Magalhães
sergio1412@gmail.com;
sergiomagalhaes@sergiomagalhaes.eng.br

[1] Department of Chemical Engineering, Rural Federal University of Rio de Janeiro, BR-467, Km7, Campus of UFRRJ, Seropedica, Rio de Janeiro 23890-000, Brazil

This paper demonstrates the preliminary results of a developed software which in a brief future could attend to an urgent need of drilling fluid operational condition control (Godhavn et al. 2011), such as wellbore pressure control in ultra-deep oil wells, or flow rate during gas invasion. This software was conceived to provide early diagnostics and interventions, which is, in this scenario, of most importance to avoid costly and even deadly drilling disasters (Oort and Brady 2011). Knowing that in drilling scenarios human resource can be expensive to maintain full time (Miller et al. 2011), the software primary objective is to automate every step needed to implement the controllers. The software was divided into three self-designed algorithms, named as automatic plant identification (API), auto-tuning (ABAP) and controllers auto-switch (CAS).

Foundations for API creation

The use of PID conventional controllers is justified due to their simplicity of design and efficiency in general industrial applications (Astrom 1985). The main problem about a PID controller is the fact that the parameters of the controller must be adjusted properly to satisfy a desired performance (Cetin and Iplikci 2015; Panda et al. 2004). In order to determine such parameters, the specialist may proceed by a phenomenological modeling of the system or follow some empirical approach. The great majorities of applications are complex to be rigorously modeled, due to its various coupled dynamics which are usually unknown. Empirical methodology does not require modeling but can be an exhausting task if performed manually. To overcome this difficulty, the proposed software automatically completes the empirical method proposed by Sundaresan and Krishnaswamy (1977). This self-designed algorithm (API) is based on a feedback response (Jeng et al. 2014) and is capable of tuning controllers in industrial processes with a fairly first-order dynamics with dead time responses. Drilling fluids hydraulic dynamics typically can be approximated to a first time order dynamic with dead time (Vega et al. 2012 and Ruiz et al. 2014). The theory behind this algorithm will be explained in "Detailing the automatic plant identification (API)" section.

Foundations for ABAP creation

Even with optimum tuned controllers, unknown disturbances or major changes in the process may occur. This may lead to a necessity of re-adjustment in the previous calculated parameters. The auto-tuning feature (self-gain schedule) is capable of an online parameters recalculation, avoiding the necessity of the entire system re-evaluation. This is important for drilling fluids controlling mostly because of their diversity with distinct physicochemical properties. If auto-tuning is off, a controller once tuned for water-based mud may fail to an oil-based mud, in example.

Some studies reported the necessity of auto-tuning techniques in order to improve performance. Soyguder et al. (2009) applied a self-tuning PID controller based on fuzzy logic to successfully control a HVAC system (heating, venting and air-conditioning). The authors demonstrated that auto-tuned PID had the best performance when compared to standard PD and PID controllers. Lim and Chatwin (1995) developed a similar research where the authors observed that auto-tuning the PID controller parameters was necessary to successfully control a CO_2 laser manufacturing system. Oh et al. (2006) presented a tuning technique for their PID strategy using genetic algorithm. The author's modeling achieved good results for a sewing commercial machine.

The auto-tuning presented by this paper was based on heuristic rules, in which a complex mathematical solution is not necessary. The heuristic rules are self-enforcing, based on system responses in order to optimize the controller performance. Reznik et al. (2000) used a similar technique in their study. The authors presented a table where rules were applied according to the response being observed in the controlled variable. The theory behind this algorithm will be explained in "Detailing the auto-tuning (ABAP)" section.

Foundations for CAS creation

Still based on Reznik's work (2000), the authors affirmed and proved experimentally that the combination between two different types of controller (PID and fuzzy controllers) can produce better results instead of using only one or another. They concluded that similarities between controllers could create a synergy that can be exploited for the creation of new control strategies. Lagerberg and Breitholtz (1997) also implemented a similar technique in an exothermic CSTR control. Similar approach was observed in Haber-Haber et al. (2007), Haber et al. (2010), Navarro-Lopéz and Cortéz (2007), Oh et al. (2004) and Shahri and Balochian (2012).

This paper also developed a similar feature, but instead of a hybrid controller, the controllers auto-switch (CAS) was developed. The CAS promotes a real-time change in controllers' parameters or even the controllers themselves in order to mitigate overshoots and damping responses. When this feature is used in combination with auto-tuning, the result is a system working in a closed loop, presenting fast response and rejecting disturbances without damping or overshoots responses.

The three modules presented were developed from reported experiences in the literature where modifications on classic PID controllers contributed to an overall improvement in process control. The main reason to merge classic control with techniques based on heuristic rules, such as Boolean and fuzzy logic, is to minimize

inefficiencies which are singular to that specific process. Therefore, this work effort was to demonstrate that petroleum and gas industries may benefit from the techniques presented here, validated from experimental data, since the literature lacks on specific studies involving drilling fluids control characteristics.

Materials and method

Drilling fluid flow loop

A simplified scheme of the automated drilling flow loop unit and its installed devices is presented in Fig. 1. The unit has a production capacity of 4000 L of water- or oil-based fluid, monitoring in real time its physicochemical properties for quality control purposes. It monitors pressure, flow rate, temperature, viscosity, density, electrical stability and conductivity, solids content, size distributions of the suspended solids, fluid loss and oil–water ratio. The software was tested in this unit not only to be validated but also to improve its performance, increasing the drilling fluid final quality.

Design of the developed software

From theories found in the literature for classical control strategies based on feedback response, the software was developed following the sequence of commands described in Fig. 2.

The code first step is the selection of the manual or automatic mode, in other words, choosing to turn on the application or not. If manual mode is selected, the manipulated variable will be controlled manually by the user, configuring an open-loop system. If automatic mode is selected, a series of requests will be demanded by the application in order to allow the automatic control, configuring a closed-loop system.

Once automatic mode is selected, the program will search for the controller's parameters. If found, the user may proceed to choose which set of parameters is going to be used: those calculated from Ziegler-Nichols (ZN) (Ziegler and Nichols 1942) or Cohen–Coon (CC) (Cohen and Coon 1953) method. Once done, the user must choose which

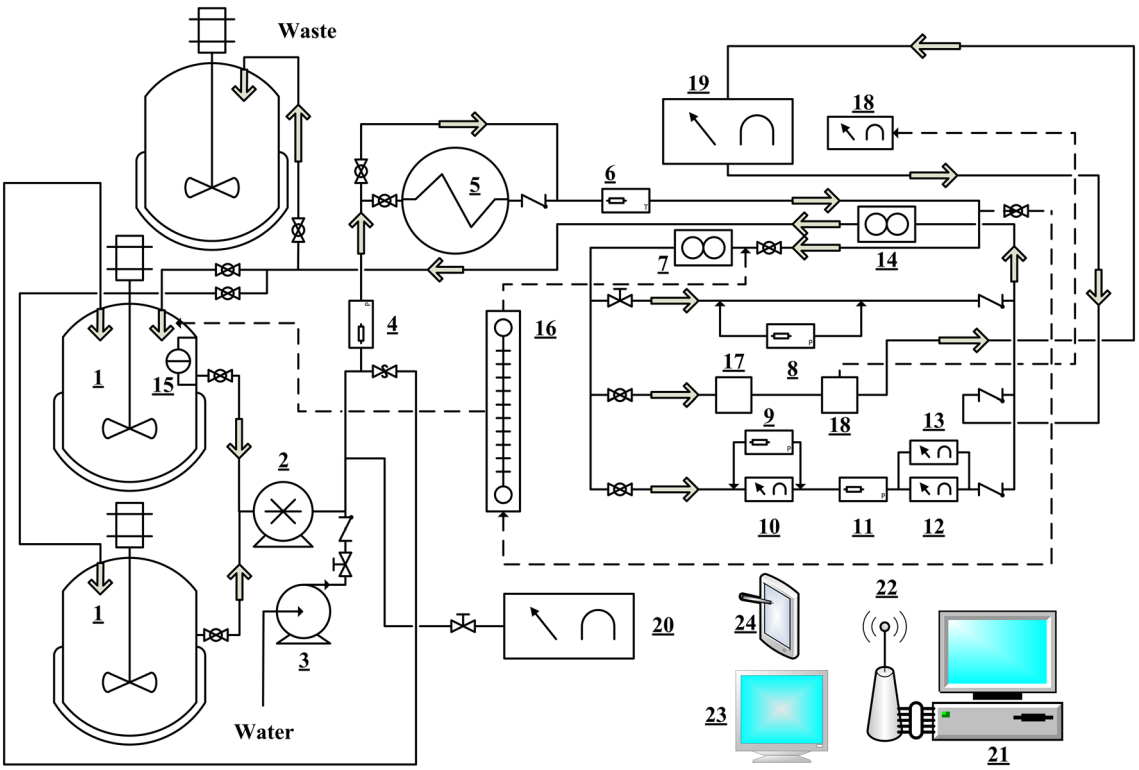

Fig. 1 Simplified schematic of the drilling fluid flow loop. 1—Stirred tanks, 2—main pump, 3—auxiliary pump, 4—pressure gauge, 5—heat exchanger, 6—thermocouple, 7—flow meter, 8—tubular viscosimeter, 9—pressure drop sensor, 10—coaxial cylinders viscosimeter, 12—electrical conductivity meter, 13—electrical stability meter, 14—density meter, 15—level meter, 16—flow in facture testes, 17—ultrasonic sensor, 18—particle size sensor, 19—water in oil sensor, 20—HTHP on line cell, 21—host computer (main HMI), 22—wireless router, 23/24—remote terminal (secondary HMI)

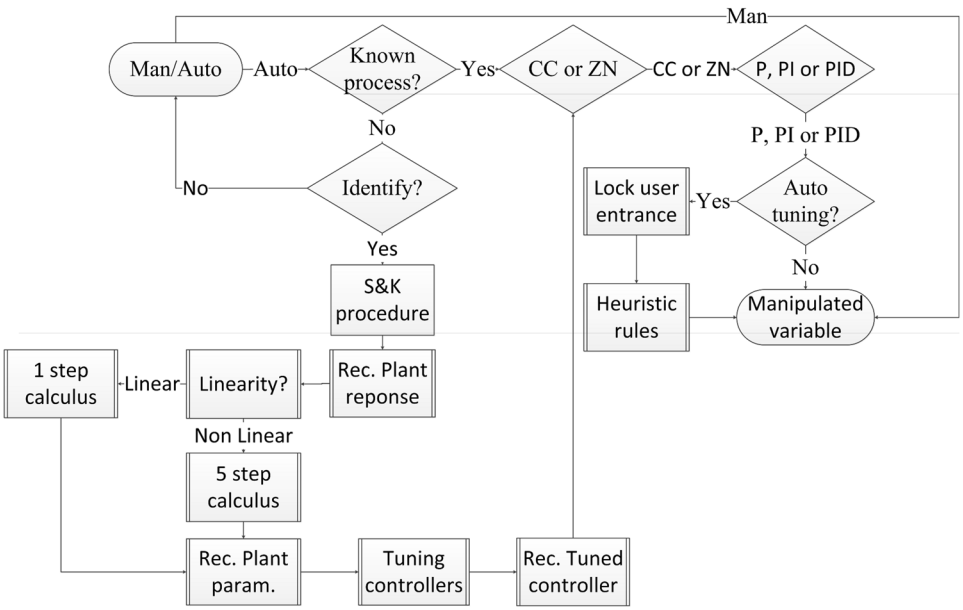

Fig. 2 Simplified schematic of the software design

controller the application should use to close the loop: P, PI or PID. The program automatically recognizes the compatible parameters previously chosen and loads into memory. After CAS configuration, one may use PI-ZN for changes in set point and PI-CC during minimization of disturbances, or P-CC until offset is reached and then triggers a change to PID-ZN.

Detailing the automatic plant identification (API)

In case of failure during search of the controller parameters, the user may proceed to identify the physical unit. This happens when the application is being installed for the first time on the computer or when the user wants to reevaluate the previously recorded parameter due to process changes. This feature was developed not only to minimize the time lost during manual calculation, but also to dismiss the presence of dedicated engineers making it accessible to lower levels of qualified manpower, which are generally at the front line of the process (Miller et al. 2011).

Before the "Identify" button is selected, the user must inform in what range the controlled variable should work. For example, it can be desired to limit the valve opening between 30 and 100% or limit the engine power between 0 and 80%.

To enlighten how the identification algorithm works, let us set an example of controlling flow rate by the pump power manipulation in a range of 0 and 80%. When the button "Identify" is selected, the system goes to 40% and 10 steps will be equally divided above and below the half point. Table 1 demonstrates the software procedures.

Once the procedure is done, the program takes less than one second to plot all the recorded data (demonstrated in Table 1) and compare the obtained responses at each step, reporting if the system response is linear or nonlinear.

Immediately after, the system proceeds to re-identify the plant accordingly to its linearity. This procedure is described in Table 2.

Table 1 Automated procedure developed based on SK technique

Stage (% of pump power)	Status
40% standing by	Not recording data
40–48% (starting identification)	Recording data
48% back to 40%	Not recording data
40–56%	Recording data
56% back to 40%	Not recording data
40–64%	Recording data
64% back to 40%	Not recording data
40–72%	Recording data
72% back to 40%	Not recording data
40–80%	Recording data
80% back to 40%	Not recording data
40–32%	Recording data
32% back to 40%	Not recording data
40–24%	Recording data
24% back to 40%	Not recording data
40–16%	Recording data
16% back to 40%	Not recording data
40–8%	Recording data
8% back to 40%	Not recording data
40–0%	Recording data
0% back to 40% (end of identification)	Not recording data
40% standing by	Not recording data

Table 2 Procedure adopted after the linearity is determined based on SK technique in open loop

Response from linearity test	Next step is…	Stage (% of pump power)	Status
Linear	1-step calculus	0% (standing by)	Waiting steady state
		0–80%	Recording data
		80% back to 0%	Not recording data
		0% (standing by)	Proceeding to calculation
Nonlinear	5-step calculus	0% (standing by)	Waiting steady state
		0–16%	Recording data
		16% back to 0%	Not recording data
		0–32%	Recording data
		32% back to 0%	Not recording data
		0–48%	Recording data
		48% back to 0%	Not recording data
		0–64%	Recording data
		64% back to 0%	Not recording data
		0–80%	Recording data
		80% back to 0%	Not recording data
		0% (standing by)	Proceeding to calculation

If the system is linear, one set of parameters is enough to control the entire range of operation. In case of nonlinearity, a 5-step calculus is made dividing the unit into 5 different parts, as shown in Table 2. Therefore, a nonlinear unit will have 5 different sets of parameters to control the entire range of operation. However, to avoid the constant parameters swap, the program uses an average weighted by the magnitude of each step.

Detailing the auto-tuning (ABAP)

Unexpected process changes may lead to the necessity of controllers' parameters revaluation. Instead of keeping re-identifying the unit every time, the controllers do not act as expected, and the ABAP feature can do minor real-time changes in the parameters.

The parameter changing is performed by heuristic rules created from the experience of drilling fluid engineers and several results available in the literature. When the feature is on, the data entry for the controllers' parameters is locked to the user and the application now is responsible for informing such values. Every change in the parameter is reversible through the "reset" button. The heurist rules are described in Fig. 3.

The algorithm was developed to search for three types of inefficiency: slow offset elimination, overshooting (or dumping) and oscillatory behavior. For slow offset elimination, the application automatically increases K_c, which increases the controller gain as well as its integral action. As a side effect, the system may become unstable with high overshooting (dumping) effects or oscillation. In case that happens, the action to take is to decrease K_c. This cycle of attempts is maintained until three recordings are made. This recording procedure prevents the system to get stuck in an endless loop. After the third record, the application freezes the K_c value and starts tuning the integral parameter.

After τ_1 is changed, the recording is restarted and the algorithm starts over again. This cycle is repeated until no inefficiency is detected, which causes the auto-tuning to enter in standby mode.

Secondary parameters exist in order to make the auto-tuning working correctly. Such parameters are required to set the offset confidence levels, range of actuation, oscillatory frequency tolerance and others. Offset confidence levels, for instance, will inform to the application, and values of K_c and τ_1 may be chosen (sets maximum and minimum possible values). This will prevent the auto-tuning feature to adjust parameters which will make the unit unstable, like $\tau_1 = 0$ or $K_c \rightarrow \infty$.

The range of actuation informs to application the tolerance to start changing parameters. For example, if tolerance is chosen as 5%, although the auto-tuning is on, if the controlled variable is distant from set point in 5% or less, the parameters will not be changed. This is fundamental as no industrial processes are capable of maintaining the controlled variable with an offset of 0%.

When the offset is minimized, the controlled variable may vary around the set point due to sensor or/and process noise. The oscillation frequency tolerance was created to distinguish between instability and natural oscillation. Usually, oscillation has a well-defined frequency, causing the controlled variable to fluctuate around set point in similar time intervals. Noises do not present such a mappable behavior.

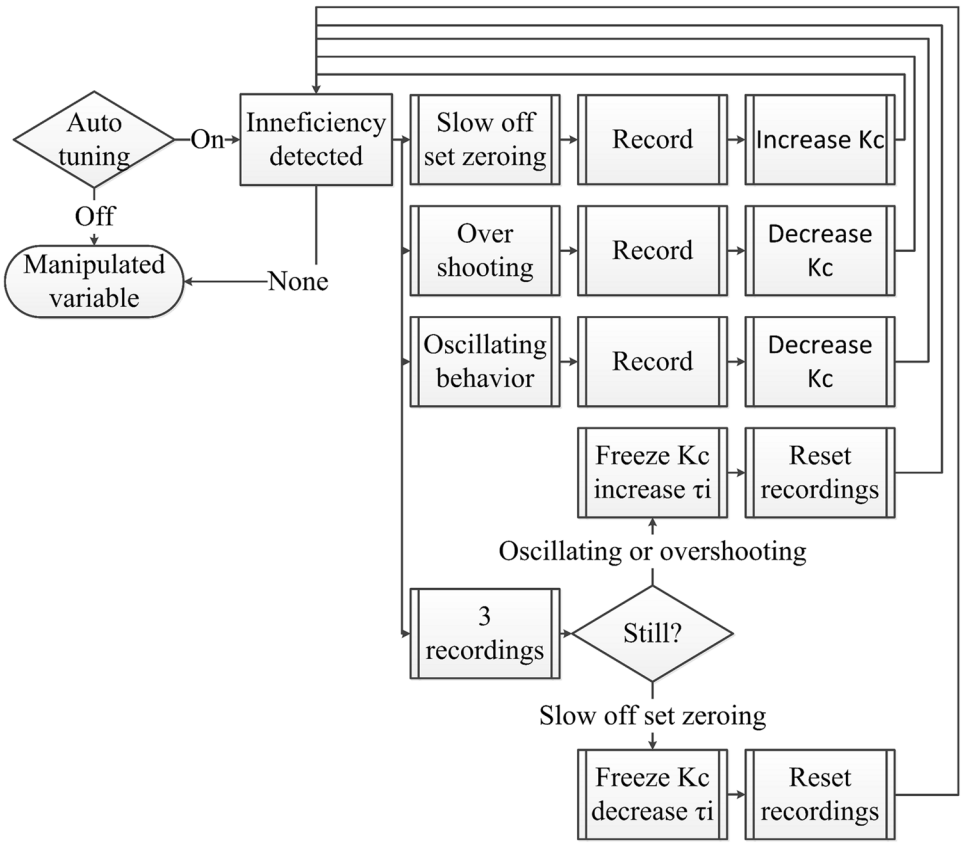

Fig. 3 Simplified schematic of the application design

Layout of the developed application

The HMI of the main application has the layout demonstrated in Fig. 4. This interface is responsible for user–system interaction.

The number caption presented in Fig. 4 is detailed in Table 3.

The main HMI is loaded on the screen when the application is executed. From it, the user is able to perform the unit identification, turning on and off the automatic control as well as the auto-tuning feature. The CAS feature is configurable only by programming. If more detailed information is required, the user may load the second HMI, which contains more technical information and the whole system algorithm. The relation between both HMI is a relation MASTER/SLAVE, where the information entered by the user in the main HMI is passed down to the secondary HMI, which performs all the tests and calculations, returning them to the main HMI. The secondary HMI image can be observed in Fig. 4.

The number caption presented in Fig. 5 can be detailed in Table 4.

Case study: controlling flow rate manipulating the pump power engine

Results obtained using the API feature

When the developed software is freshly installed in Windows environment, no unit and controller parameters are recorded in database. Therefore, the only action possible is to use the API feature. Figure 6 demonstrates how the application commanded the pump engine and how the flow rate was recorded while API was in action.

In Fig. 6, both charts of engine pump power and flow rate are aligned in time. As illustrated in "Detailing the automatic plant identification (API)" section and Table 1, the unit was divided into 5 forward and 5 downward steps, as can be observed during the first 600 s.

The unit was in manual mode at approximately 29% of power when the "Identify" button was pressed. From this point, the pump power engine was automatically raised to the middle point of the 5 steps, which was 50% due the power limit set as 100%. The response of the flow rate for each power step can be observed in black.

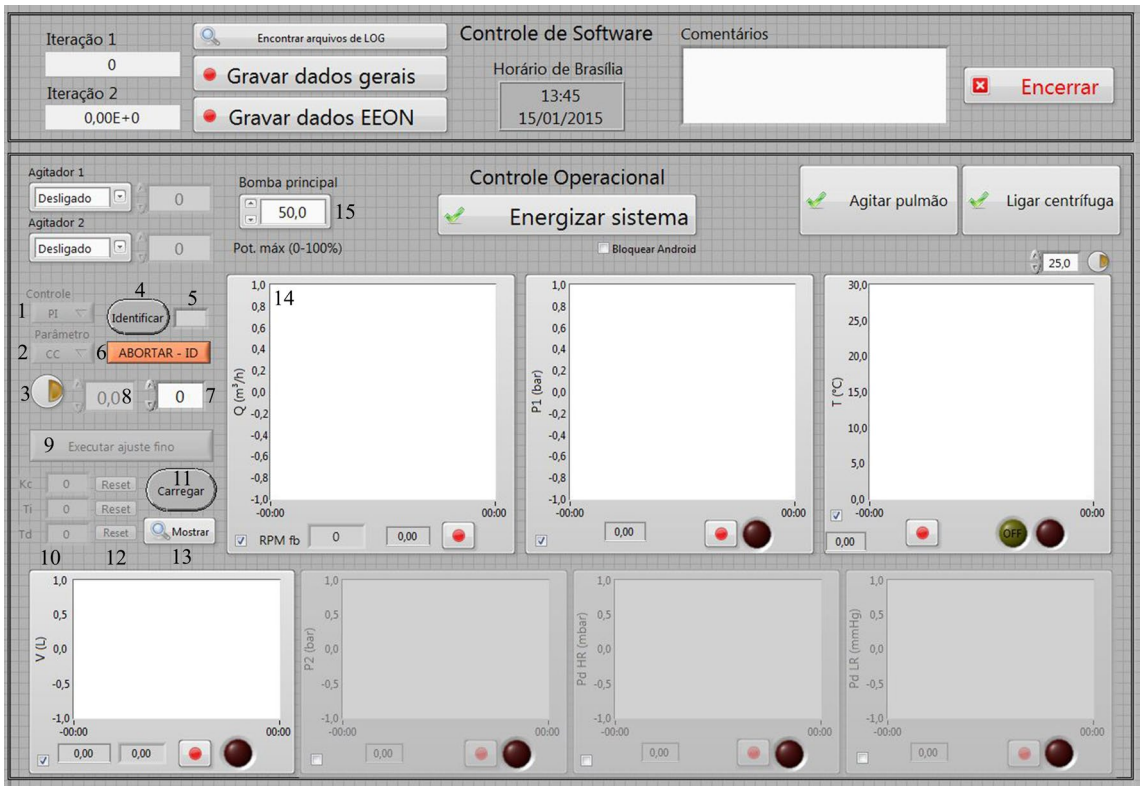

Fig. 4 Photograph of the main HMI interface

Table 3 Caption of Fig. 4

Number	Description
1	To choose the type of controller
2	To choose parameters calculated by CC or ZN theory
3	To turn on automatic control (closed loop)
4	To start plant identification
5	Indicator of plant linearity
6	To abort plant identification (return to manual)
7	To inform set point
8	To inform pump power (manual mode)
9	To turn on auto-tuning
10	Indicator of the values of the parameters being used by the controller
11	Enforce parameters load into memory
12	Reset parameters to its original value (ZN or CC)
13	Load slave HMI to check the history of all parameters calculation
14	Graphic indicator of the set point and controlled variable behavior
15	Range limitation during identification test

The procedure took approximately 600 s to be done. At this stage, the application started to compare the responses magnitudes with equal engine power step. In the case presented in Fig. 6, the system dynamic was nonlinear.

The next step was to automatically calculate the unit parameters dividing the plant into 5 steps forward (see "Detailing the automatic plant identification (API)" section and Table 2). These steps can be seen in Fig. 6 after 650 s of test.

The five steps forward were performed in approximately 300 s, and then, the system was switched back to manual mode. Figure 7 presents the typical screen where results are presented.

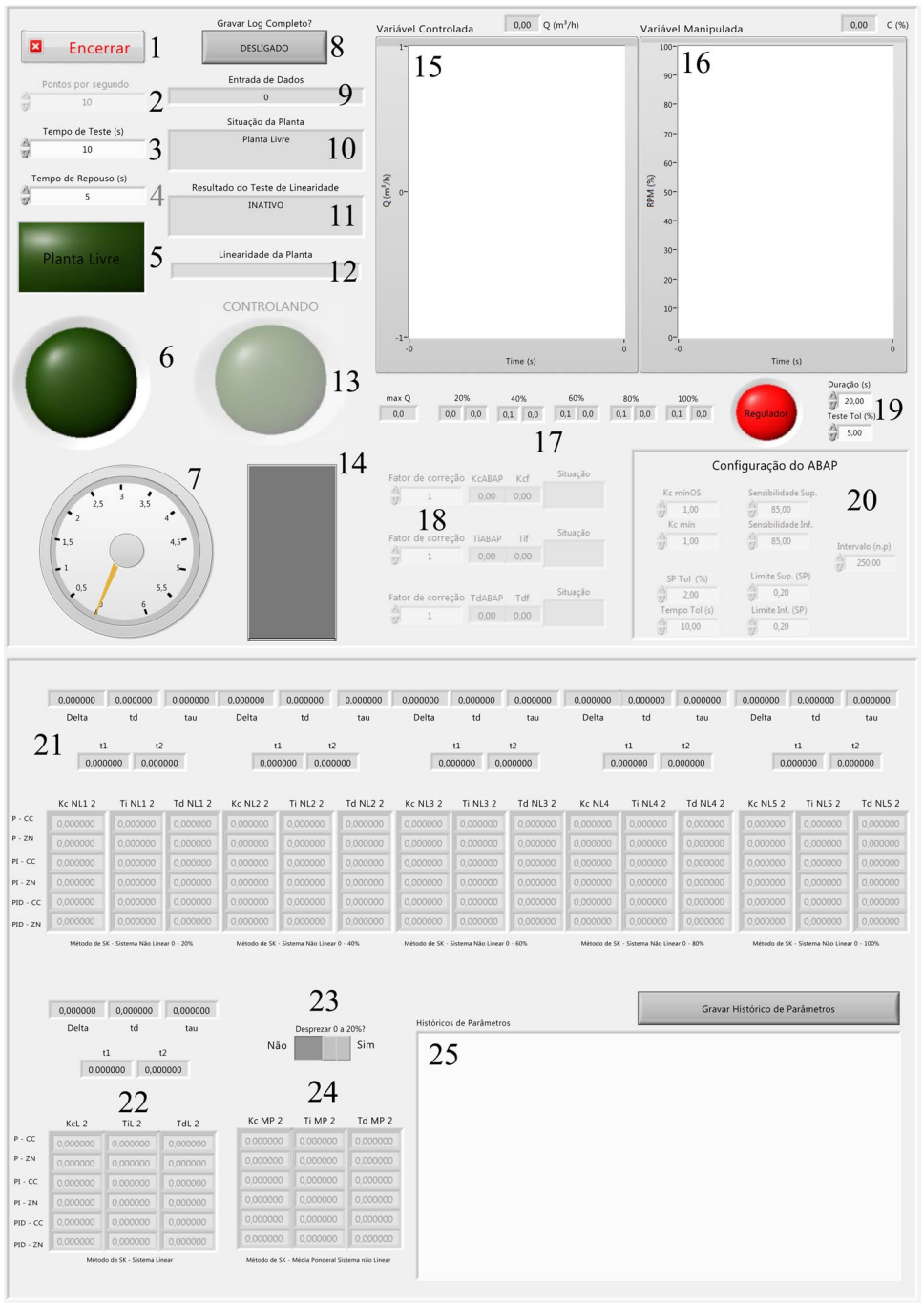

Fig. 5 Photograph of the slave HMI interface

The five plant parameters (*Delta*, *td*, *tau*, *t1* and *t2)* were reported for each step done. Also, all the combinations between the control strategy and its tuning method can be observed: P-CC, P-ZN, PI-CC, PI-ZN, PID-CC and PID-ZN.

At the bottom of Fig. 7, the weighted averages used to tune the controllers were presented. The dimmed region would be used only if the unit dynamic was linear. The whole process took no more than 16 min to be completed.

Results of the controller's performance calculated by API feature

Figure 8 presents the comparison between PI-CC and PI-ZN performances. PI controllers were chosen due to its popularity in petroleum industries (Ruiz et al. 2014).

It can be observed in Fig. 8 that the controller with CC parameters (in blue) was faster when compared to ZN parameters (in black), when set point variation (in red)

Table 4 Caption of Fig. 5

Number	Description
1	To terminate slave HMI (this will lead to manual operation only)
2	To change rate of communication between software and hardware
3	Parameter to determine when the system should record data
4	Parameter to determine when the system should not record data
5	Indicator of plant usability (standby or during plant identification)
6	To indicate if application is recording data or not
7	To indicate the state of a second variable (i.e., system pressure)
8	To record every value on the screen
9	To inform how many times the system has interacted with hardware
10	To indicate what the application is doing (standby, identification, etc.)
11	To indicate the subroutine during plant identification
12	To inform what was the result of the linearity test
13	To visual inform if the plant is operation in closed or open loop
14	To visual indicate the actual state of the manipulated variable (in %)
15	To graphically indicate the history of the controlled variable
16	To graphically indicate the history of the manipulated variable
17	To indicate how the system was divided during the 5-step calculus
18	To indicate the proceeding being done by the auto-tune feature
19	To set up the tool which detects servo and regulator events
20	To regulate the set of parameters needed for the auto-tuning feature
21	To indicate the history of all calculated parameters
22	Parameters actually being used in linear system
23	To give an option to discard the first set of parameters (5-step calculus)
24	Parameters actually being used in nonlinear system
25	To enter and record general comments

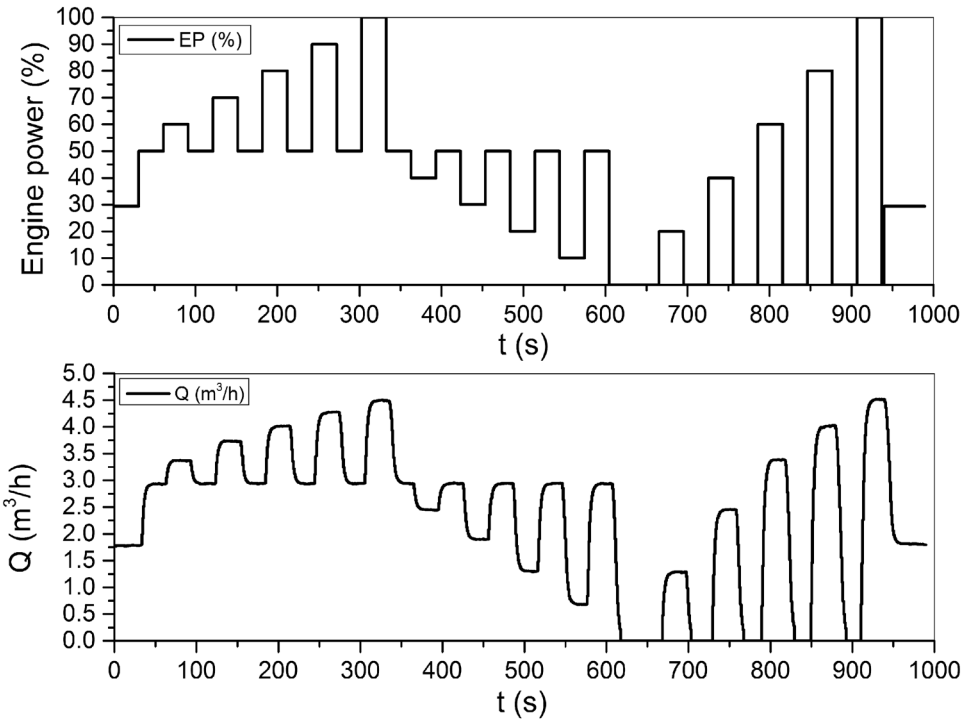

Fig. 6 Flow rate and engine power in function of test time

Fig. 7 Typical calculated parameters for a nonlinear system

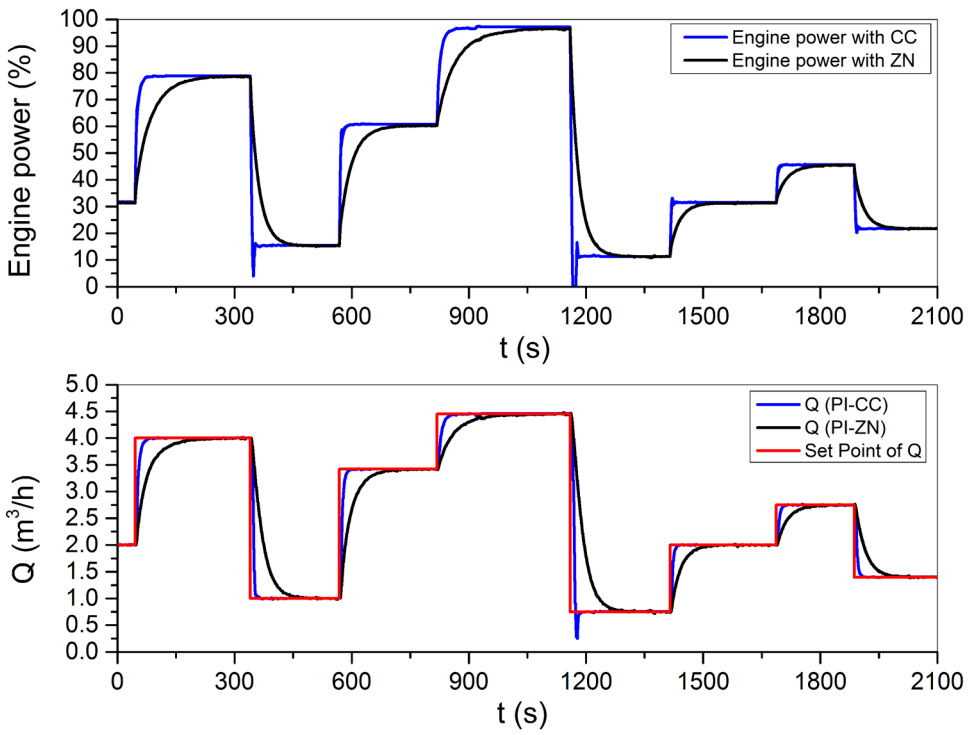

Fig. 8 Flow rate and pump power in function of time during test of setting point manipulation

was imposed. This was expected because integral action is greater in CC parameters than in ZN parameters. Similar results were observed experimentally by GirirajKumar et al. (2010).

As shown in Fig. 8, it can be seen that CC parameters were 15.16 and 3.39, for K_c and τ_1, respectively, while for ZN the parameters were 13.49 and 12.05. The integral action for CC was almost 4 times greater than that found for ZN. High integral action contributes for a faster response, but also may cause instability in some scenarios, which happened when an abrupt decrease on set point was requested. When set point was at 4.45 m^3/h and changed to 0.75 m^3/h, PI-CC exhibited a damping response, causing flow rate to go under the set point. This behavior was not observed when PI-ZN was applied.

Results using the auto-switch option

The auto-switch option was developed specifically to improve performance in scenarios where overshoots or dumping responses are not accepted. Once the regions where such behaviors are identified, the user may choose to switch the controller or its parameters. Figure 9 demonstrates the results of the tests where a switch from PI-CC to PI-ZN was made after two concomitant situations occurred: Variation on set point was greater or equal to -3.7 m^3/h, and controlled variable was at 10% or less from it. After the variable has been controlled within a tolerance of 2% around the set point, the application switched back to PI-CC, ready to change it again if necessary.

Figure 9 shows the comparison of the three controllers performance: PI-CC, PI-ZN and PI-CC/ZN/CC (which means PI-CC than PI-ZN than PI-CC). It was verified by the results that the overall performance of the auto-switch feature (in blue) was better than the other controllers, as it was as fast as PI-CC and did not cause dumping, such as PI-ZN.

Results obtained using the ABAP feature

The last self-design feature tested was the auto-tuning (ABAP). An experiment was performed where several set point variations were inputted, in order to compare the fastest controller (PI-CC) with the PI-CC auto-tuned. The results are presented in Fig. 10.

It can be seen in Fig. 10 that the performances of both controllers were similar in the first set point change (blue and black). After the second set point change, the auto-tuning feature, which has learned from the last change, started to act. From that time forward, the auto-tuned PI-CC performance was better than the standard one.

Disturbance rejection tests were also performed to prove the auto-tuning feature importance in extreme and non-predictable situations. Figure 11 presents the results comparing the standard and the auto-tuned PI-CC.

The first disturbance observed in Fig. 11 was derived by a valve opening, which decreased the flow rate, simulating a leak or even an inadequate valve manipulation. The

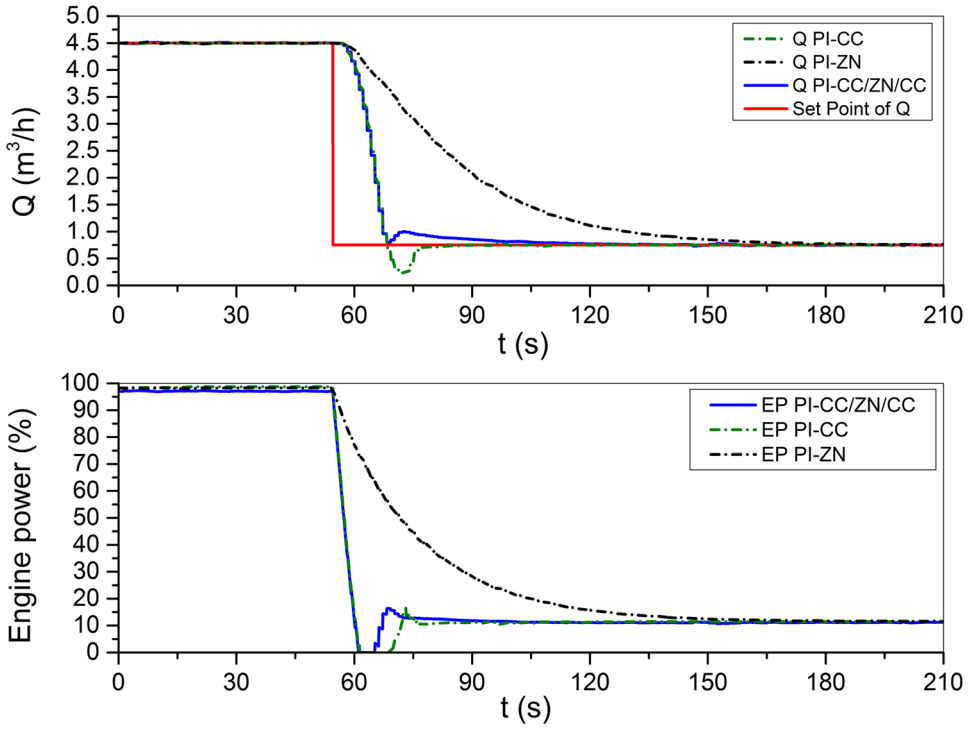

Fig. 9 Flow rate and pump power in function of time during test with auto-switch option on

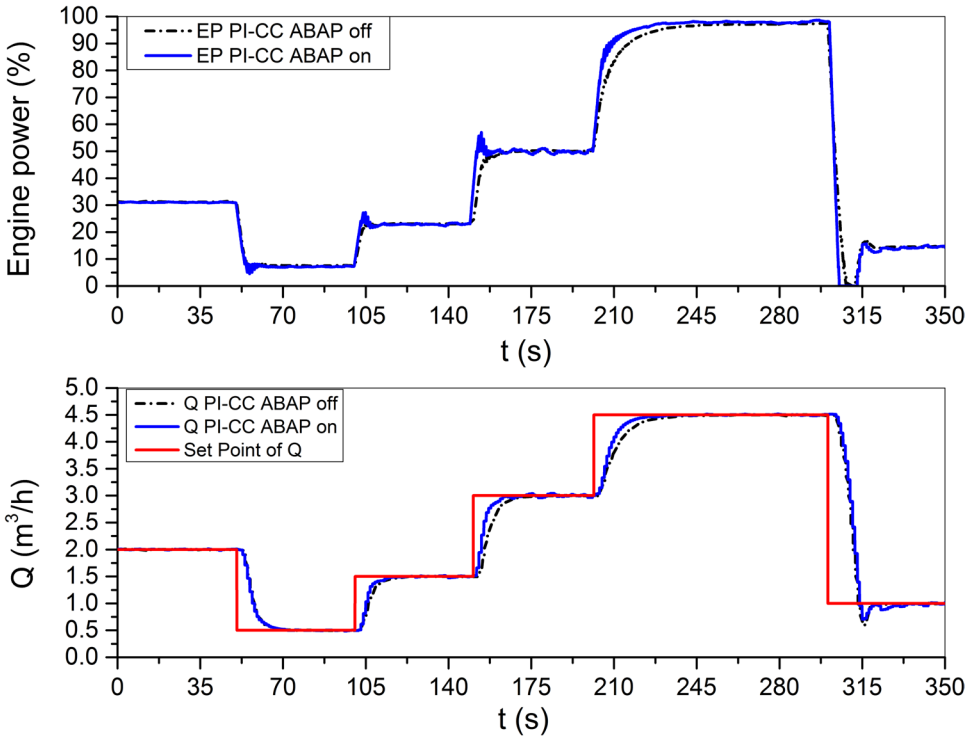

Fig. 10 Flow rate and pump power in function of time during test of variation of set point with auto-tuning feature on/off

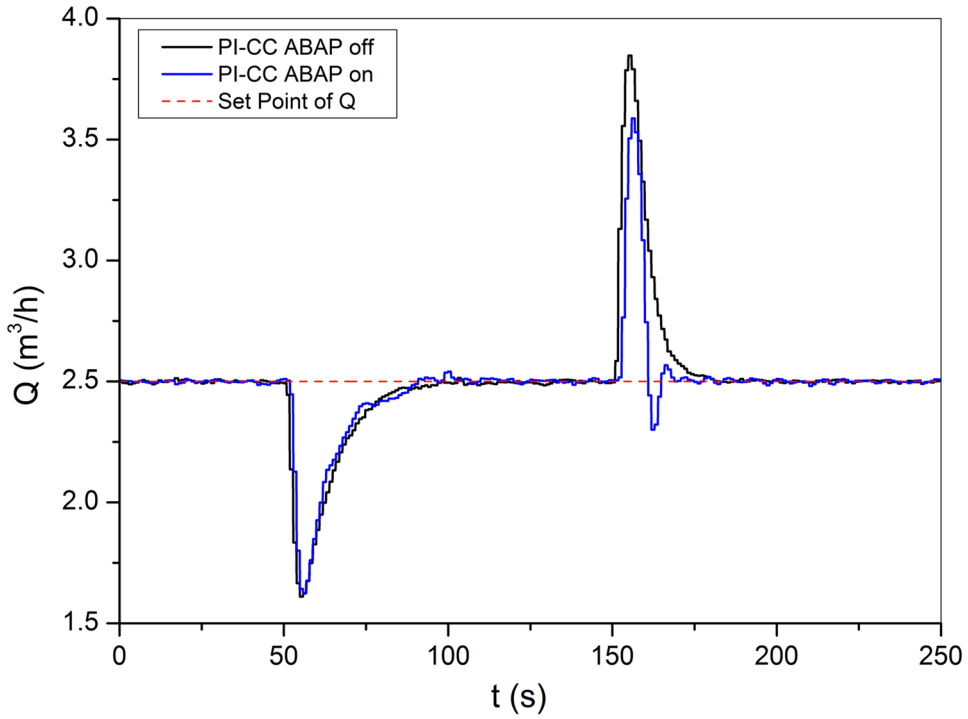

Fig. 11 Flow rate and pump power in function of time during test of disturbance rejection with auto-tuning feature on/off

controllers actuated and the flow rate were reestablished to its set point. The second disturbance was the closing of the same valve, which can represent an undesirable influx or an inadequate valve manipulation. Due to the auto-tuning feature, the second disturbance was quickly controlled when compared to standard PI-CC. The flow rate peak was better handled.

Two extremes can be reached, depending on the ABAP feature regulation parameters sensitivity. If it is set to be less sensitive, there will be no difference between auto-tuning

and standard PI-CC. If it is set to very sensitive, all process changes will be handled as fast as possible at a cost of a possible oscillation behavior. The limits of the controller's parameters changing will be formed automatically. In case of a lower sensitiveness, no changes will be made. In case of high sensitiveness, the K_c parameter will be raised to a point where oscillation behavior will occur. At this point, the auto-tuning rules group will mitigate it as soon as possible, up to a maximum accepted value. In practical manners, the more the auto-tuning is used, the more it converges to an average optimum set of parameters. So, if there are no process changes, the auto-tuning will not actuate. While the opposite is true, as more "changes" are experienced by the tool, more prepared it will be to mitigate future similar disturbances.

Figure 12 shows an experimental result where high sensitiveness was chosen and an oscillatory behavior was observed (CAS was kept off to demonstrate the ABAP performance alone). The test results demonstrated how fast the oscillation was mitigated and also that this tool is not only an optimizer, but also a guarantee that the controllers will never lead to instability, contributing to process safety.

Figure 12 presents a standard PI-CC and the auto-tuned PI-CC performances, and similarity was found during the first process change. After the second change, the auto-tuned control was slightly faster, but with a dumping response. This behavior was registered and K_c was again regulated to avoid a future dumping. It can be seen that auto-tuned PI-CC was slightly faster on the next two set point changes. After that, an unknown disturbance occurred leading to major changes in K_c (due to the sensitiveness choice), which allowed the controller to be considerably faster than standard PI-CC. For the next disturbance, an oscillatory behavior was detected. The oscillation was quickly mitigated by reducing K_c to lower levels, as described in the algorithm explained in "Detailing the auto-tuning (ABAP)" section. To prove that new K_c was better than the one found for standard PI-CC, a new disturbance was done not only changing the flow rate set point, but also introducing a simultaneous leakage. With these two new disturbances, the auto-tuned PI-CC proved to be faster than PI-CC. ABAP not only optimizes the controller, but also prevents instability.

Conclusion

This paper described a self-adaptive application developed to process control based on feedback response with a minimum manpower involvement. The software has been developed to attend the petroleum industries and its flow and pressure control, but due to its robustness, any industrial process which behaves like a first-order dynamic system may benefit from it. The main application features are the automatic plant identification (API), the auto-tuning (ABAP) and the controllers auto-switch (CAS).

Preliminary tests were performed on a drilling fluid production unit. API feature identified and calculated successfully the linearity and the controller's parameters of the drilling fluid production unit. The ABAP tuned on real time the parameters of the fastest controller, reducing its time response and increasing its acting velocity. This tool, with

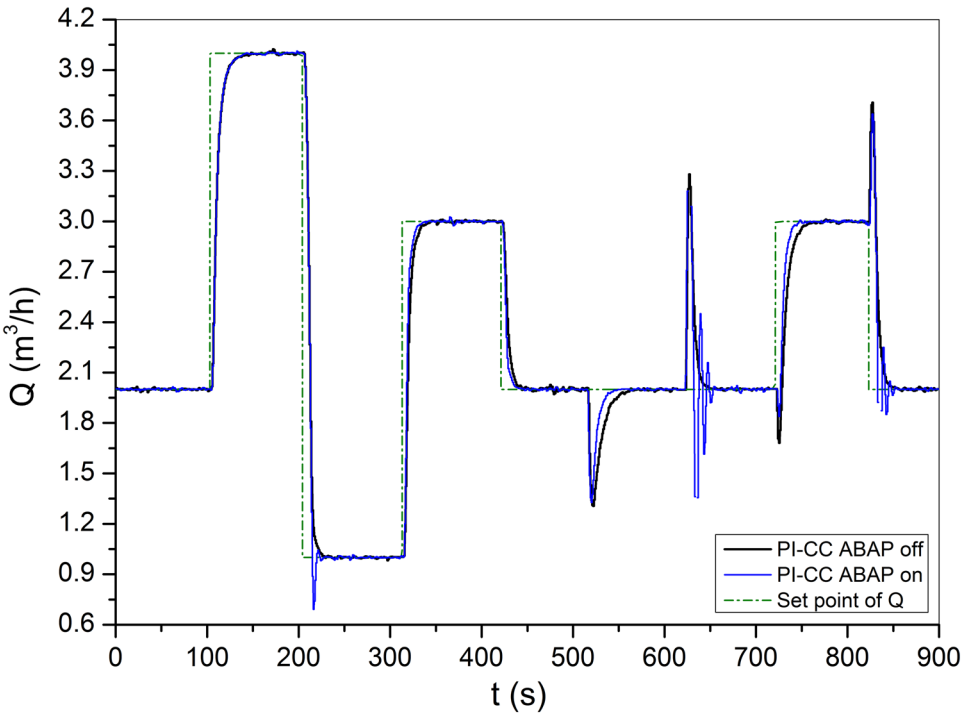

Fig. 12 Flow rate and pump power in function of time during a test with ABAB configured as sensitive and CAS turned off

the CAS feature, proved to be capable of mitigating oscillatory, overshooting and dumping behaviors, increasing profitability and safety for the drilling fluid production unit.

The three self-designed features combined turned the application into a powerful tool with its high automation level and minimum requirement for human interaction. This application is adequate for petroleum industries, where a robust process controller is always needed and full-time dedicated engineers may be expensive to maintain. With this scenario, the presented automated software finds its importance.

Acknowledgements The authors appreciate the financial support offered by CENPES (PETROBRAS Research Center) (4600293210) and give thanks to the scientific cooperation provided by CAPES and PPGEQ/UFRRJ.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

Astrom KJ (1985) Process control—past, present and future. IEEE Control Syst Mag 6(4):3–9

Bourgoyne AT Jr, Millheim KK, Chenevert ME, Young FS Jr (1991) Applied drilling engineering. Second Printing Society of Petroleum Engineers

Broussard S, Gonzalez P, Murphy R, Marvel C (2010) Making real-time fluid decision with real-time fluid data at the rig site. Society of Petroleum Engineering (SPE). SPE Drilling Conference and Exhibition, Abu Dhabi, UAE. SPE 137999

Caenn R, Chillingar GV (1996) Drilling fluids: state of the art. J Pet Sci Eng 14(3):221–230

Cetin M, Iplikci S (2015) A novel auto-tuning PID control mechanism for nonlinear systems. ISA Trans 58:292–308

Cohen GH, Coon GA (1953) Theoretical considerations of retarded control. Trans ASME 75:827–834

Gandelman RA, Martins AL, Teixeira GT, Aragão AFL, Neto RMC, Lins DGM, Lenz C, Guilardi P, Mari A (2013) Real time drilling data diagnosis implemented in deepwater wells—a reality. OTC-24275-MSOTC-24275-MS, Rio de Janeiro, October 29th to 31th, Brazil

GirirajKumar SM, Jayaraji D, Kishan AR (2010) PSO based tuning of a PID controller for a high performance drilling machine. Int J Comput Appl (0975-8887) 1(19):12–18

Godhavn J-N, Pavlov A, Kaasa G-O, Rolland NL (2011) Drilling seeking automatic control solutions. Preprints of the 18th IFAC World Congress, Milano (Italy)

Haber RE, del Toro RM, Gajate A (2010) Optimal fuzzy control system using the cross-entropy method. A case study of a drilling process. Inf Sci 180:2777–2792

Haber-Haber R, Haber R, Schimittdiel M, del Toro RM (2007) A classic solution for the control of a high-performance drilling process. Int J Mach Tools Manuf 47:2290–2297

Jeng J-C, Tseng W-L, Chiu M-S (2014) A one-step tuning method for PID controllers with robustness specification using plant step-response data. Chem Eng Res Des 92:545–558

Lagerberg A, Breitholtz C (1997) A study of gain scheduling control applied to an exothermic CSTR. Chem Eng Technol 20(7):435–444

Lim SY, Chatwin CR (1995) A PID-knowledge-based system for nonlinear process control. Integr Comput Aided Eng 2(4):291–298

Magalhães SC, Scheid CM, Calçada LA, Folsta M, Martins AL, Sá CHM (2014) Development of on-line sensor for automated measurement of drilling fluid properties. IADC/SPE-167978-MS

Miller A, Minton RC, Colquhoun R, Ketchion M (2011) The continuous measurement and recording of drilling fluid density and viscosity. SPE/IADC Drilling Conference and Exhibition, Amsterdam, Netherlands. SPE/IADC 140324

Navarro-Lopéz EM, Cortés D (2007) Sliding-mode control of a multi-DOF oilwell drillstring with stick-slip oscillations. In: Proceedings of the 2007 American Control Conference, Marriott Marquis Hotel at Times Square, New York City, USA

Oh YT, Kwon WT, Chu CN (2004) Drilling torque control using spindle motor current and its effect on tool wear. Int J Adv Manuf Technol 24:327–334

Oh T-S, Lee W-H, Kim I-H (2006) Parameter optimization of 2-DOF-PID controller using genetic algorithm. Int J Appl Electromagnet Mech 24:131–145

Oort EV, Brady K (2011) Case-based reasoning system predicts twist-off in Louisiana well based on Mideast analog. Special Focus–Drilling Technology

Panda RC, Yu C, Huang H (2004) PID tuning rules for SOPDT systems: review and some new results. ISA Trans 43:282–295

Reznik L, Ghanayem O, Bourmistrov A (2000) PID plus fuzzy controller structures as a design base for industrial applications. Eng Appl Artif Intell 13(4):419–430

Ruiz A, Jiménez JE, Sánchez J, Dormido S (2014) A practical tuning methodology for event-based PI control. J Process Control 24:278–295

Shahri ME, Balochian S (2012) Fractional PID controller for a high performance drilling machine. Adv Mech Eng Appl (AMEA) 2(4):232–235

Soyguder S, Karakose M, Alli H (2009) Design and simulation of self-tuning PID-type fuzzy adaptive control for an expert HVAC system. Expert Syst Appl 36:4566–4573

Sundaresan KR, Krishnaswany PR (1977) Estimation of time delay, time constant parameters in time, frequency and Laplace domains. Can J Chem Eng 56:257–262

Vega MP, Freitas MG, Folsta MG, Gandelman RA, Martins AL (2012) Experimental control of annulus pressure while drilling oil wells. In: Proceedings of the 2012 IFAC workshop on automatic control in offshore oil and gas production, Norwegian, University of Science and Technology, Trondheim, Norway

Ziegler JB, Nichols NB (1942) Optimum settings for automatic controllers. ASME Trans 64:759–768

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.