Available online at www.sciencedirect.com

SCIENCE DIRECT®

PERGAMON

Control Engineering Practice 11 (2003) 1469–1480

CONTROL ENGINEERING PRACTICE

www.elsevier.com/locate/conengprac

Sintolab: the REPSOL-YPF PID tuning tool

R. González-Martín[a], I. López[b], F. Morilla[b], R. Pastor[b,*]

[a] Dpto. de Control Avanzado, PETRONOR, Apdo. 1418, Bilbao 48080, Spain
[b] Dpto. de Informática y Automática, UNED, Avda. Senda del Rey 9, Madrid 28040, Spain

Received 31 October 2002; accepted 28 March 2003

Abstract

This paper describes a tuning tool for PID controllers. What makes this tool different to others available in the market is: (a) its ability to model the most important load disturbances, (b) its wide range of tuning methods that cover two types of specifications (features of time responses and features of frequency responses), (c) its possibilities in order to import or export data and to determine the more adequate control parameters exportable to the distributed control system. The calculated MISO model enables an improved tuning of the PID loop, both for the servo and regulatory cases. This is of special interest in the Petrochemical Industry where most of the PID controllers have a fixed design criterion, load disturbances rejection or setpoint following, like in model predictive control applications. This tool, which combines direct experience in the industry with the solid theoretical knowledge of a group of university researchers, is already being widely used in all the REPSOL-YPF refineries and some of its petrochemical industrial complexes.

Keywords: Distributed control systems; Industrial control; PID tuning; Process parameter estimation; Software tools

1. Introduction and motivation

Over the last decade, an extraordinary amount of effort, i.e. money, has been invested in the petrochemical and refining industry to support the development of advanced control applications. The main reason has been the maximisation of the production margin and/or profit. The equation *money makes money* is the underlining principle.

It is the advanced control engineer's role to translate the company's overall economics objectives into specific targets, which should be eventually carried out by conventional controllers. Most of the times, the controller is a PID standard algorithm.

Whenever the process complexity or the control target becomes more complicated, a more advanced multivariable control strategy might be required. This is the case of model predictive control (MPC) applications (Qin & Badgwell, 1997), where the higher hierarchy controller is generating the setpoints to the PID controllers. This situation is shown in Fig. 1.

The question, which the engineer has to face is of the type: *Do I need a more complicated controller, or just a PID controller will be enough*? Unfortunately, there is not a straightforward answer to this question, not even an easy way to calculate the PID controller performance bounds.

It would be very useful to have a procedure to determine when a more advanced controller is required. This is achieved by evaluating the PID performance under a certain set of conditions. These conditions are defined by the process dynamics and the load disturbances characterisation. Obviously, the PID performance is closely related to its parameters: K_P, T_I and T_D. Therefore, the PID tuning procedure plays a key role in determining whether a PID controller is able to achieve the design criterion for that particular loop or not.

Two scenarios are possible. The best scenario occurs when the tuned PID controllers are able to do the job properly meeting all the design criteria. Alternatively the PID performance may not be good enough, thus requiring a more advanced controller. Though the PID

*Corresponding author. Tel.: +34-913988383; fax: +34-913986697.
E-mail addresses: rgonzalezm@repsolypf.com (R. González-Martín), ilopez@dia.uned.es (I. López), fmorilla@dia.uned.es (F. Morilla), rpastor@dia.uned.es (R. Pastor).

0967-0661/03/$ - see front matter © 2003 Elsevier Science Ltd. All rights reserved.
doi:10.1016/S0967-0661(03)00082-0

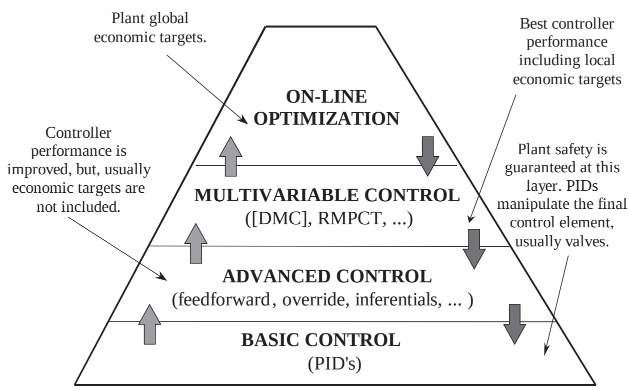

Fig. 1. Typical hierarchical control structure.

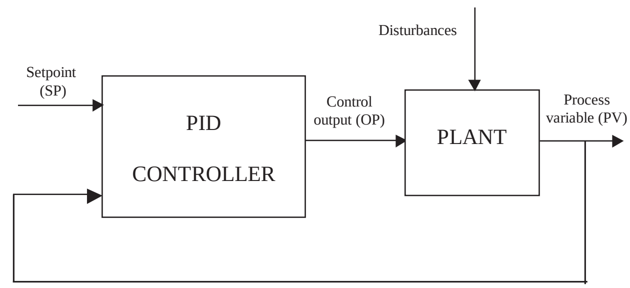

Fig. 2. Basic structure of a PID loop in *Sintolab*.

controllers will keep operating, their design criteria will be less demanding.

At the beginning of 1994 this was a big concern for REPSOL-PETROLEO and PETRONOR S.A. (companies belonging to REPSOL-YPF group), and the marketplace did not then offer a product that fulfilled all the requirements. At that time, no product addressed the issue of modelling disturbances, so, it was decided to develop a proprietary tool called *Sintolab*. The tool has been developed by a team composed of people from the UNED University and from the advanced control departments of REPSOL-YPF. The tool has continuously evolved since then, now being one of the essential tools necessary for the setting up of any MPC project.

Section 2 lists *Sintolab*'s main features. Section 3 is dedicated to the project management involved when tuning a PID loop. In Sections 4 and 5 the most important issues of *Sintolab* are emphasised. Two case studies are presented in Section 6.

2. Sintolab features

Sintolab is a 32 bits MS-Windows™ application (Morilla & Pastor, 2001; Pastor, 2001) for the tuning and simulation of PID control loops. The basic control structure of these loops is shown in Fig. 2, where:

The *Plant* stands for the dynamic behaviour of the process to be controlled. It includes relationships from the control output signal (OP) and disturbances to the process variable (PV).

The *PID CONTROLLER* stands for the element that calculates the control output (OP) based on the setpoint (SP) and the process variable (PV). Usually this block reproduces a commercial PID controller algorithm. The most common PID controller used in REPSOL-YPF is performed by the Honeywell's TDC-3000™ distributed control system.

Sintolab includes:

- Tools to manipulate and process historical data in the form of vectors. *Sintolab* is able to import data from the distributed control system and ASCII files in multiple formats.
- Procedures to estimate and validate the plant model based on test data. The model includes the dynamics of the most important measured disturbances.
- Model generation based on first principles for level processes.
- A wide range of tuning methods and different interfaces which vary depending on the final user. *Sintolab* supports two different user profiles: the control engineer and the instrument technician.
- A simulation environment which allows the user: (a) to study plant response to predefined inputs, (b) to study the closed loop response to setpoint, load or disturbances changes, (c) to evaluate the controller's performance under similar conditions to those present in the actual process.
- Control parameters validation based on the characteristics of the controller where the PID is implemented, e.g. limits checking according to type of equation, etc.
- Automatic tuning reports generation. This is extremely useful to keep track of all the work done in a particular loop over time. The history of the loop is invaluable information in the tuning process.

3. Project management

Sintolab's projects involve, in its wider sense, a collection of objects used to determine the control parameters of the main object, named *Controller*. Only a single main object is included in each project, but a project can have as many basic and auxiliary objects as it is needed. Some of these objects are described below as well as the actions required to generate them.

3.1. Main object

The controller is the main object of a *Sintolab* project; it includes all the information about the module that is used in the distributed control system, and its relative situation within the plant. This controller, see Fig. 2, reproduces the ideal PID algorithm control (Aström &

Table 1
Equations of the ideal PID control algorithm

PID	$OP = K_P\left(1+\frac{1}{T_I s}+\frac{T_D s}{\alpha T_D s+1}\right)(SPN - PVN)$
PI-D	$OP = K_P\left(1+\frac{1}{T_I s}\right)(SPN - PVN) - K_P\frac{T_D s}{\alpha T_D s+1}PVN$
I-PD	$OP = K_P\frac{1}{T_I s}(SPN - PVN) - K_P\left(1+\frac{T_D s}{\alpha T_D s+1}\right)PVN$

Hägglund, 1995), that is considered the standard implementation by ISA (The Instrumentation, Systems, and Automation Society). Three equations (PID, PI-D, I-PD), shown in Table 1, are allowed, with the following control parameters: K_P proportional gain, T_I integral time constant, T_D derivative time constant and α filter factor on the derivative action (fixed to 0.1).

The controller block in *Sintolab* also has other characteristics that influence the tuning methods and the closed loop simulations:

- It updates the control output (OP) in regular intervals of time, the sampling time (in seconds), which is chosen by the user.
- The control action can be direct or reverse. Equations in Table 1 are formulated considering that the control action is reverse and that, in addition to PV, SP and OP, another two normalised signals (PVN and SPN) are in the control loop. See Fig. 3.
- It incorporates a reset anti-windup mechanism so the control output signal (OP) always has values between the maximum and minimum fixed by the user.

3.2. First level objects

In Sintolab the following four types of basic objects are defined: vector, model, controller setting and simulation.

Vector. Vectors are data containers. These data may be plant data imported from the distributed control system (they will be used to estimate models) or any data available in the project, which has been generated by simulations or by processing a vector.

Model. Models are representations of the plant in *Sintolab*. They are generally obtained starting from real data during the estimation procedure and they are the basis for the control parameters calculation.

The plant model, see Fig. 3, is composed of a process model block and as many disturbance model blocks as measurable disturbances have been recorded. *Sintolab* contemplates three types of parametric models for the process, as shown in Table 2. These models are typically representative of the industrial process dynamics. First principle dynamic level models are also treated in Sintolab as a special case of the third model (model for integrating processes), with K as function of the tank's geometric properties, $T_1 = 0$ and $T_0 = 0$.

Controller setting. A controller setting object represents a possible set of parameters (K_P, T_I and T_D) for the controller block. The control parameters calculation in *Sintolab* can be carried out in seven different ways.

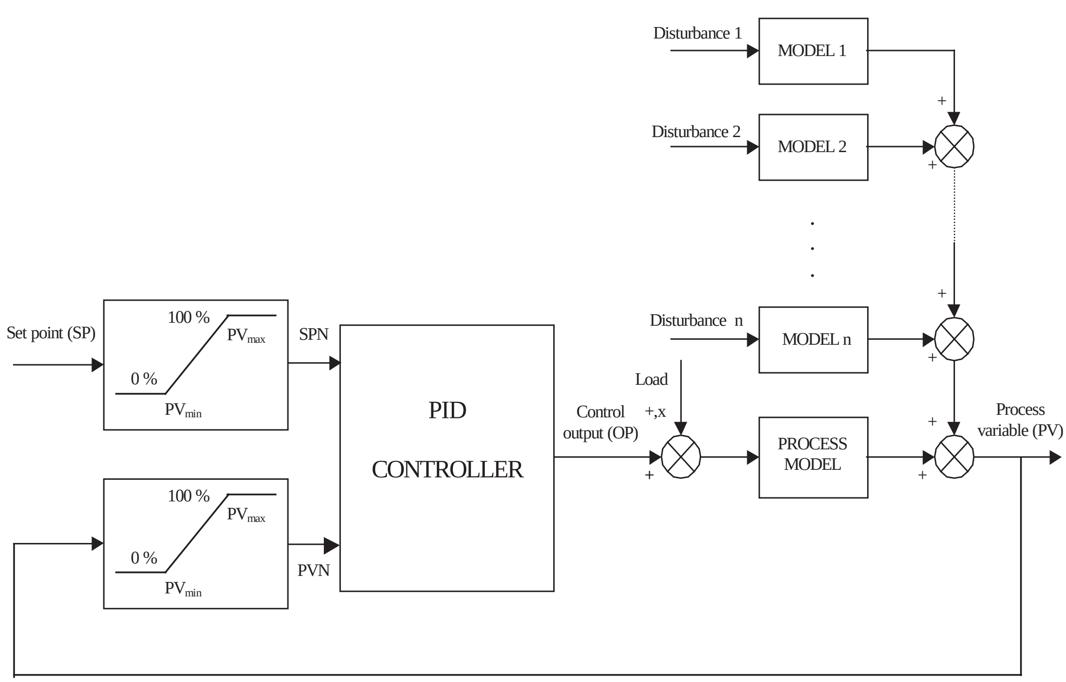

Fig. 3. Control loop model used in *Sintolab*.

Table 2
Parametric models

Type of model	Transfer function
First-order plus dead time	$\dfrac{K}{(T_1 s+1)} e^{-T_0 s}$
Second-order plus dead time	$\dfrac{K}{(T_1 s+1)(T_2 s+1)} e^{-T_0 s}$
First-order plus dead time and integrator	$\dfrac{1}{s}\dfrac{K}{(T_1 s+1)} e^{-T_0 s}$

These methods are summarised in Table 3, along with its corresponding specifications as determined by the user.

Simulation. Simulations are experiments performed with a model or a controller setting from the project. Each experiment will have its own duration and input signals determined by the user.

3.3. Second level objects

Sintolab also provides additional objects in order to help the control engineer during the project, these are: vector list, data set and model structure.

Vector list. It is an object that groups vectors, making it possible to view them simultaneously. It can be used to visually validate models or controller setting, to compare simulations of the control loop in different situations or different controller settings under the same experimental conditions. Fig. 4 shows an example of vector list with two vectors.

Data set. It is an object that groups points of a vector; the control engineer can define several regions of a vector using data sets. For example, it is possible to select one data set (according to its adequate excitation) in the estimation of the model and another data set to validate it. Data sets can be also used to create new vectors by joining different data sets or deleting invalid data sets from a vector. The bright and dark rectangles in the bottom of Fig. 4 show that there are two data sets in the vector list.

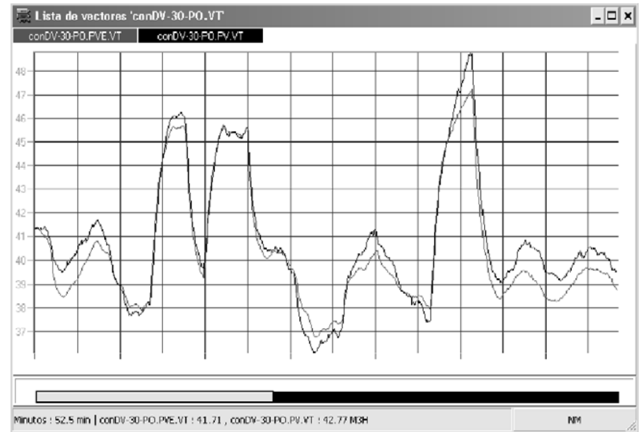

Fig. 4. Vector list of two vectors with two data sets.

Model structure. It is an object that groups several models under a common root. Whenever the dynamics of the plant (process and disturbances) are uncertain, a model structure can be used to set off several estimations followed by the selection of the most adequate model from the models generated.

3.4. Sintolab workflow

Each project in *Sintolab* is built by making a series of actions in a predefined sequence. In so doing, certain objects are used and others are generated (see Fig. 5). The main actions involved (in the correct sequence on the left side of Fig. 5) are as follows:

Controller definition consists in the description of the module used in the distributed control system, which maps its characteristics into the controller object.

Data collection represents the collection of data coming from the plant in the vectors: PV, OP and measurable disturbances (if there were any). This action

Table 3
Tuning methods

Method	Specifications
Tuning by phase margin	The desired phase margin and the associated frequency.
Tuning by gain margin	The desired gain margin and the associated frequency.
Tuning by phase and gain margin	The desired phase and gain margins (Morilla & Dormido, 2000).
Tuning by settling time and/or maximum overshoot	The desired settling time and/or maximum overshoot for setpoint, load or disturbance change. Also, due to the iterative nature of this method, it's possible to specify initial values for control parameters.
Robust tuning	Settling time, overshoot and control parameters ranges. Grid size for control parameters is required.
Tight control	No specifications are needed. Only is valid for level models.
Average control	The deviation level allowed and the maximum flow disturbance expected. Only is valid for level models.

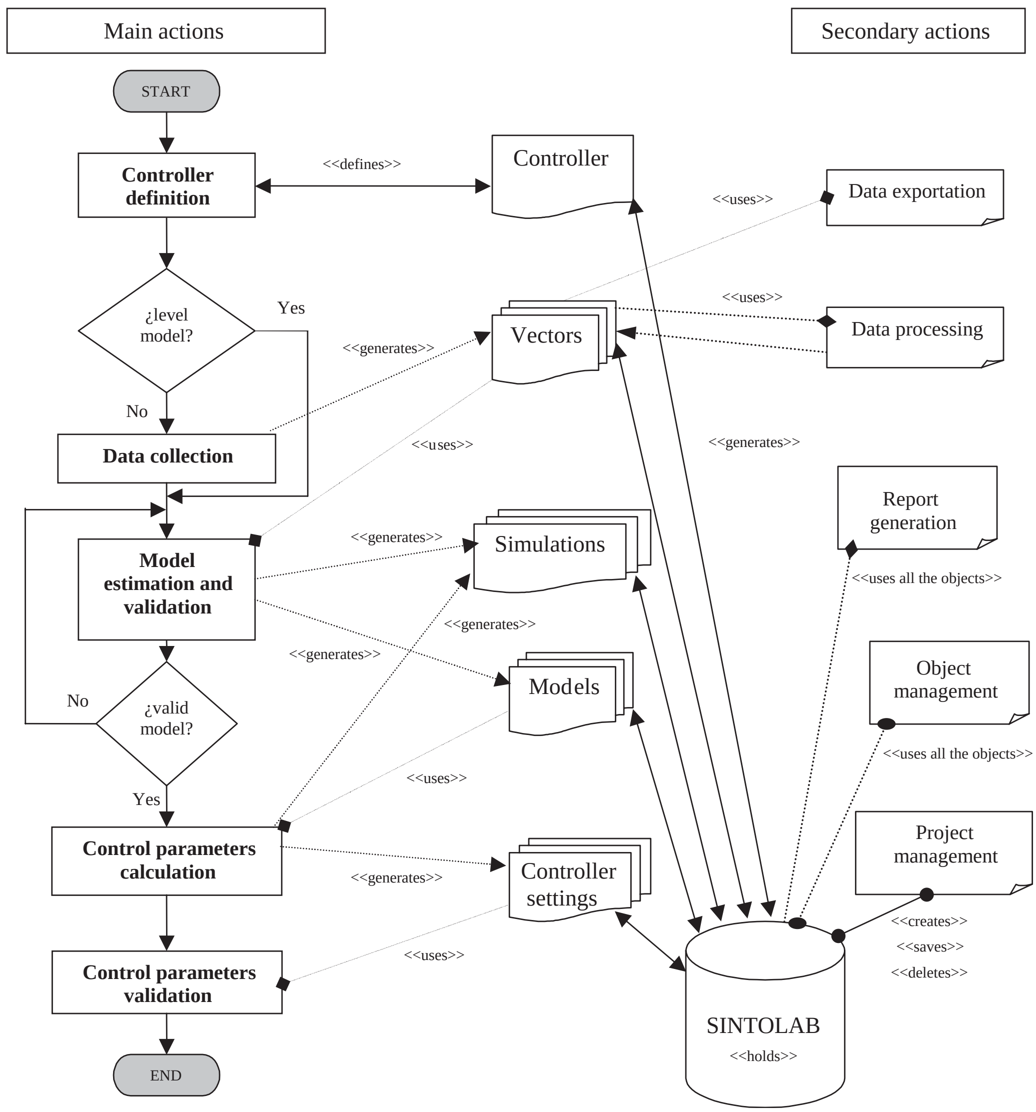

Fig. 5. *Sintolab* workflow.

is not necessary for level tuning based on the geometry of a vessel.

Model estimation and validation produces a representative model plant, using previously collected data. Level models are an exception of this procedure, where the estimation is based on the tank's geometric properties instead. Further details of the estimation action are given in Section 4.

Control parameters calculation determines the PID control parameters which, together with the process and disturbances models, meet the user's specifications. In Section 5 more details of the tuning procedure are given.

Control parameters validation, which is the final action, allows the user to check whether the control parameters are exportable to the distributed control system module or not. In this action a conversion of control parameters can also be implicit (Aström & Hägglund, 1995). It takes place when the module in the distributed control system implements the interactive PID control algorithm, expressed as

$$\mathrm{OP} = K'_p\left(1 + \frac{1}{T'_I s}\right)\left(1 + \frac{T'_D s}{\alpha' T'_D s + 1}\right) \times (\mathrm{SPN} - \mathrm{PVN}).$$

In addition to the main actions, there are other secondary actions (on the right-hand side of Fig. 5), which are also important in *Sintolab*. These actions are:

Project management, with which it is possible to create, save and delete *Sintolab* projects.

Object management, with which it is possible: (a) to copy and delete objects, (b) to inspect the properties of any object, (c) to compare several objects of the same type. These actions have been highly appreciated by *Sintolab*'s users, because they open a lot of possibilities in order to achieve the main goal, which is "to determine

Table 4
Types of data processing techniques

Type	Process	Parameters
Outlier filter	Delete outliers using an asymmetric median filter.	None
First-order filter	Filter data using a first-order discrete filter.	Constant of the filter
First-order filter with phase elimination	Filter data using a first-order discrete filter with phase elimination.	Constant of the filter
Linear interpolation	Linear interpolation between two time instants defined for a data set no valid.	Data set no valid
Smoothing	Data smooth using a simple minimum least square method.	None
Re-sampling	Re-sampling of a data vector using an integer factor of its sampling time.	Sampling time factor.
Linear transformation	Transform the data vector using the formula ($aV + b$), where V is the vector and a, b are scalars.	a, b scalars.

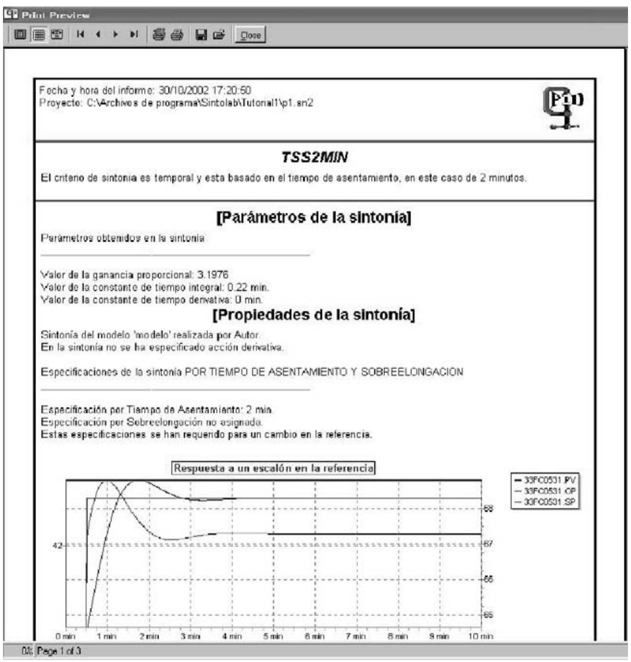

Fig. 6. Report of a PI controller tuning by settling time of 2 min for a setpoint change.

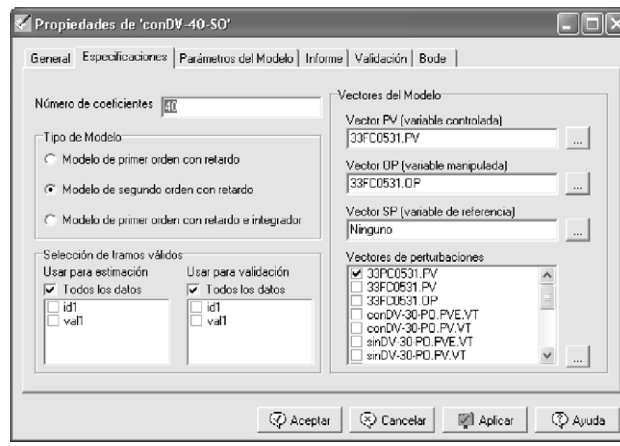

Fig. 7. Specifications for process and disturbance models estimation.

the more adequate control parameters exportable to the distributed control system''.

Data processing. Sometimes it is necessary to manipulate the vectors previous to the estimation process. An example of this happens when the data has noisy values that can be deleted performing an outlier removal operation. Table 4 shows the types of data processing techniques included in *Sintolab*.

Data exportation. The data contained in vectors can be exported to files using two different formats: the CSV data format (from Microsoft Excel©) or the ASCII representation (that can be read directly from an advanced calculus tool like Matlab©). Thus, virtually any application can use data from *Sintolab* and vice versa.

Report generation produces a description of any object used in a *Sintolab* project. Fig. 6 shows the description of a controller setting. By combining several of these reports, the user can call up a record of the steps followed until the final decision and can always answer the question: Why and how have the control loop parameters been changed? The reports may be printed or saved into files. So, it is possible to preview a report generated and saved in *Sintolab* even though the object does no longer exist.

4. Modelling details

Whenever the user requests an estimation, he must specify the number of coefficients needed to reconstruct the step response, and the type of process model (parametric) to be fitted (see Fig. 7).

This request effects a totally automated procedure based on three steps:

- *Step 1: Check the excitation*. If the register of a particular control loop contains all the signals (OP, disturbances and PV) but one of them is constant, *Sintolab* will only estimate parameters for those blocks (process model and disturbances models) where it has found valid experimental data. *Sintolab* performs the same procedure whenever it detects an input signal which only slightly influences the output. This might happen due to the small amplitude of its changes, or because the frequency content of the input signal is of little significant for the bandwidth of the process.

- *Step 2: Reconstruction of the step responses (coefficient models)*. The objective is to estimate the plant step responses using input data without any predefined pattern (Ogunnaike & Ray, 1994). All the responses will have the same number of coefficients (values) specified by the user.
- *Step 3: Estimation of the model parameters*. It consists of estimating the parameters of the process and perturbation models that better approximate the step responses that have been already reconstructed. The method of moments (Aström & Hägglund, 1995; Ogunnaike & Ray, 1994) is used for this purpose.

The estimation in *Sintolab* incorporates a final stage: the presentation of the results (text, numbers and graphics), enabling the user to validate the model. The first validation must be a visual inspection of every step response, including reconstructed responses (coefficient models) and those which have been generated using the parametric models. The final and decisive validation encompasses the complete model of the plant. It consists of graph comparisons between the real plant response and the model response, using either the same experimental data that was used to estimate the model or another set of data altogether. This validation is complemented with a numerical result: the $R2$ factor (Moore & McCabe, 1993). The closer $R2$ is to one, the better the estimation will be. For example, in Fig. 8 the model named *conDV-40-S0* is a good model given that its $R2$ factor is 0.96. Additionally, the square error sum between the plant response and the model response is also calculated by *Sintolab*, where a small figure represents a good estimate.

Each estimation in *Sintolab* logs messages as a way of facilitating the validation of the model. These messages can also be very helpful to determine whether the model is not a good one or does not satisfy the user's expectation.

Modelo	R2	Coeficientes	Tipo	K	T1 (min)	T2 (min)	T0 (min)
sinDV-10-PO	0.7	10	FOPDT	0.207276	0.3		0
sinDV-10-SO	0.7	10	SOPDT	0.207276	0.15	0.15	0
sinDV-20-PO	0.77	20	FOPDT	0.31969	0.59		0
sinDV-20-SO	0.78	20	SOPDT	0.31969	0.22	0.36	0
sinDV-30-PO	0.8	30	FOPDT	0.371396	0.79		0
sinDV-30-SO	0.8	30	SOPDT	0.371396	0.79	0	0
sinDV-40-PO	0.81	40	FOPDT	0.392524	0.9		0
sinDV-40-SO	0.81	40	SOPDT	0.392524	0.9	0	0
conDV-10-PO	0.92	10	FOPDT	0.248061	0.3		0
conDV-10-SO	0.92	10	SOPDT	0.248061	0.15	0.15	0
conDV-20-PO	0.95	20	FOPDT	0.349401	0.55		0
conDV-20-SO	0.95	20	SOPDT	0.349401	0.19	0.36	0
conDV-30-PO	0.96	30	FOPDT	0.375382	0.68		0
conDV-30-SO	0.96	30	SOPDT	0.375382	0.68	0	0
conDV-40-PO	0.96	40	FOPDT	0.37711	0.72		0
conDV-40-SO	0.96	40	SOPDT	0.37711	0.72	0	0
modelo	0.96	30	FOPDT	0.375382	0.68		0
modelo_val	0.92	30	FOPDT	0.368335	0.49		0.21

Fig. 8. $R2$ factor comparison.

5. Tuning details

When the user requests a controller setting, she must specify: what type of controller (PI or PID) she prefers, the process model to be used for the calculations, the tuning design criterion, and the corresponding specifications. See Table 3. For example, in the case of tuning by settling time and/or maximum overshoot, these specifications must be entered as shown in Fig. 9. Requesting a controller setting launches a semiautomatic procedure, so the user's intervention is required.

The calculation of the control parameters usually has as the final result a unique set (K_P, T_I and T_D), although, internally the following five situations are possible:

1. *The solution is not feasible*. This may be the case in the tuning by phase or gain margin, and it is usually due to a bad selection of the design frequency or because of the specifications being too demanding. *Sintolab* provides an interface for the selection of the design frequency in order to avoid this situation. This problem might also appear in the robust tuning due to an over specification of the grid size.
2. *The solution is unique*. This situation is common in average control and tight control, and also in tuning by phase margin or gain margin if the user specifies a correct design frequency.
3. *The solution is the best that Sintolab can offer*. This is typical of tuning methods with iterative processes: tuning by phase and gain margin and tuning by settling time and maximum overshoot.
4. *The solution is not unique, but it is automatically selected.* When the user requests *Sintolab* to look for possible solutions, it automatically selects one of them and forgets the rest. An example of this appears in tuning by phase or gain margin, when the user has specified a number of frequencies inside the set of possible solutions. In this case the control parameters with maximum integral gain are selected automatically (Morilla & Dormido, 2000).
5. *There are several solutions*, because the user has requested *Sintolab* to provide all possible solutions. They are presented to the user in a well-organised way. It is important to keep in mind that any individual solution may be picked up from then on depending on the user needs. This is a specific characteristic of robust tuning, of which Fig. 10 shows an example. The graph on the bottom left distributes the possible solutions over the T_D range of values. As there is a greater number of solutions when $T_D = 0$, it does seem more suitable to use the PI controller, although in other cases it is better to use the PID controller.

Fig. 9. Specifications for a tuning by settling time and/or maximum overshoot.

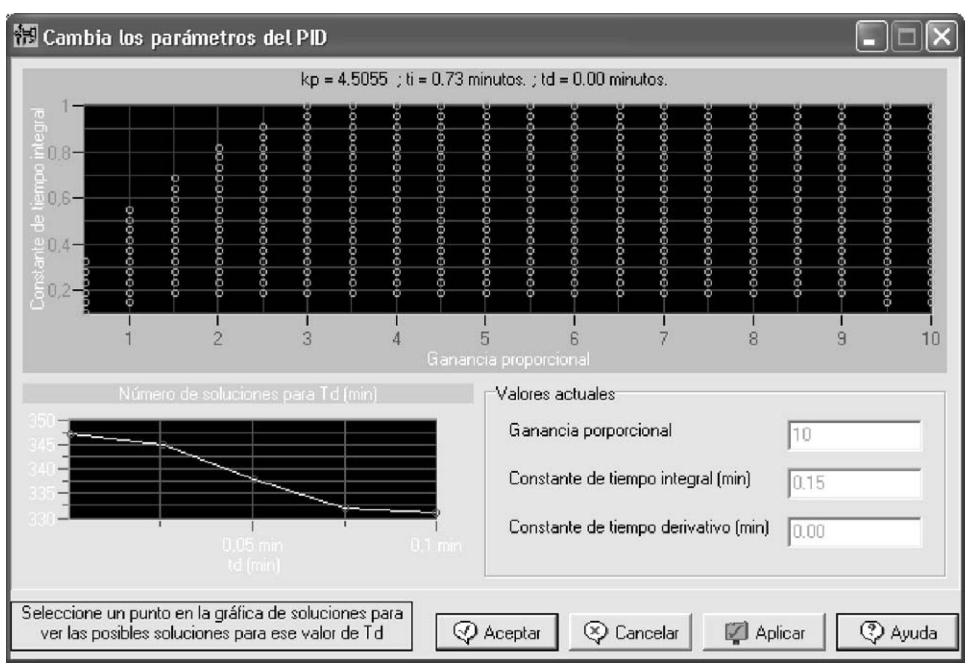

Fig. 10. Robust tuning solutions window.

An appropriate presentation of the results (reports, numbers, figures) to the user is a key point for validating the control parameters calculation. The first validation must consist of a simple check of the control parameters. The second, that is the final validation (although not always possible), is based on the analysis of the temporal and frequency response characteristics. The user can browse through to check whether the specifications are fulfilled and determine the convenience of using those control parameters or not. Among others, *Sintolab* calculates the PM (phase margin), GM (gain margin), t_s (settling time), m_p (maximum overshoot).

Only tunings with iterative processes, i.e. tuning by phase and gain margin and tuning by settling time and maximum overshoot, have an associated textual message. The message will show whether the quality index, which measures the fulfillment of the specifications, is inside the tolerance levels or, else, whether the calculation of the control parameters has stopped because the algorithm is trapped in a local minimum or has exhausted the number of iterations.

6. Case studies

This section describes two case studies, a temperature control and a level control, which clearly show the advantages of modelling the main process disturbances present at the time the test is conducted.

The model parameters and the final controller settings have been omitted to preserve confidentiality. Data shown on trend graphs has been biased for the same reason.

6.1. Reboiler temperature control

This case study presents a PID control loop with tuning difficulties. It is a reboiler outlet temperature controller from a distillation tower (TC-1). Fig. 11 shows the process flow diagram. The controller manipulates fuel-gas flow to control the temperature. Temperature is manipulated by a MPC controller (Qin & Badgwell, 1997), thus, a good setpoint following is required. This is difficult to achieve because there is a big load disturbance present in the process.

Using fuel-gas as the main combustible in a petrochemical plant is quite painful, because it is usually a residual product from other processes; the fuel-gas heat capacity very often varies, thus, affecting the outlet temperature. This makes it very difficult to establish a consistent baseline in order to start the process test. The analyser AI-1 measures this disturbance. In this case, the design criterion also specify quick disturbance rejection.

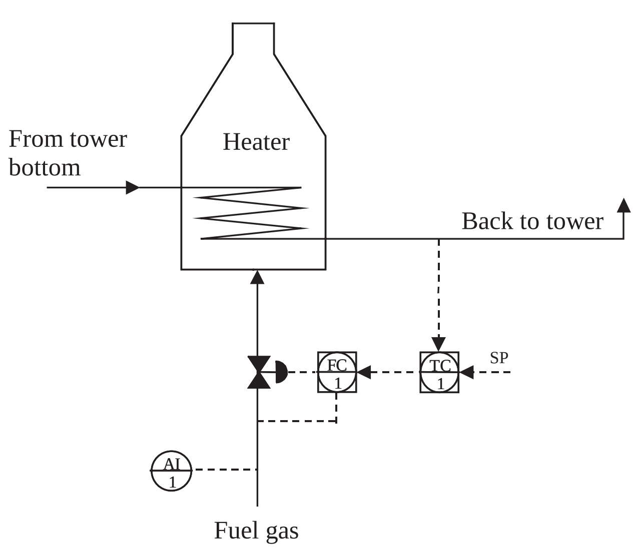

Fig. 11. Reboiler outlet temperature PID control.

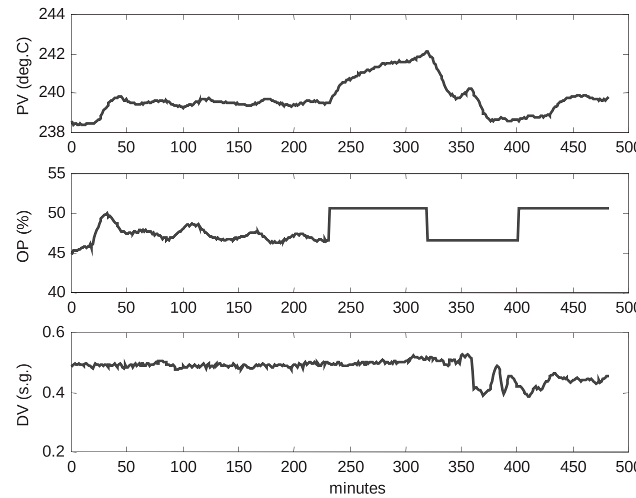

Fig. 12. Test data on the reboiler.

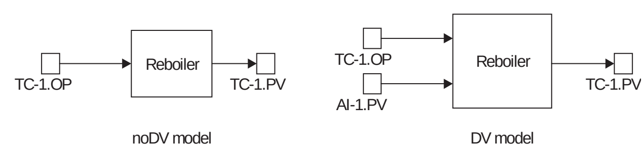

Fig. 13. Reboiler models used.

Fig. 12 shows the test carried out in the plant. The loop was set to manual mode at minute 220. A quickstep test was applied to the TC-1.OP. The DV curve shows the variations in the fuel gas heat capacity during the test.

The test data is then used to calculate the process model required to tune the loop. Two reboiler models are considered, see Fig. 13. The first (referred as noDV) includes only the TC-1.OP as an independent variable, whereas the second (referred as DV) includes the load disturbance AI-1.PV as the second independent variable.

Fig. 14 shows the step responses (coefficient models) estimated by *Sintolab*. In the upper diagram it can be clearly observed that, although the steady state gain of the models are very much the same, their dynamics are slightly different, thus, affecting the control parameters calculation. It is important to consider at this point, that the ability to model the load disturbance can help us substantially whenever a more complex control strategy is required. For example, if a feed-forward control is required, the disturbance model can be used to design the lead lag net.

Fig. 15 shows the estimated temperature vs. the real one. The estimate takes into account changes both in the controller output and fuel gas heat capacity (DV model). The goodness of the estimate is visually apparent in the graph. Though not shown here, the residuals are uncorrelated, and so the model is validated.

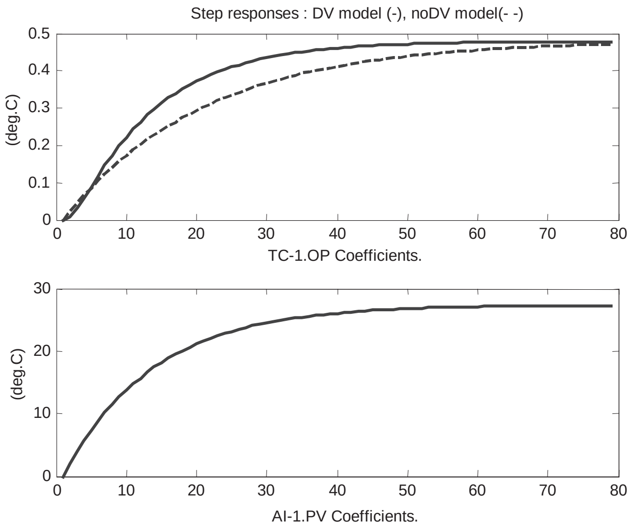

Fig. 14. Reboiler's temperature dynamic models.

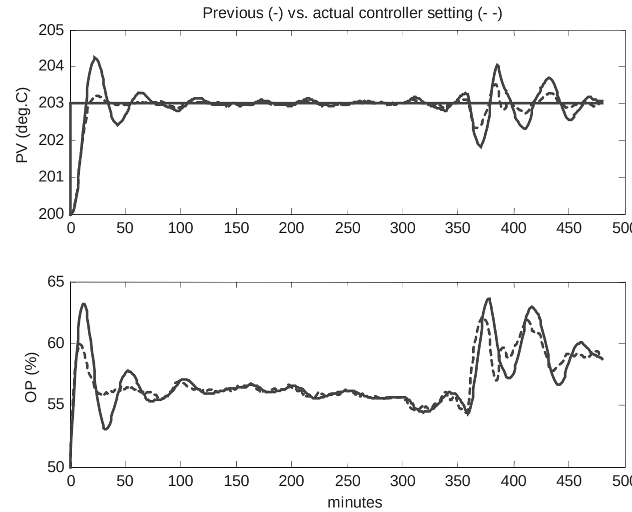

Fig. 16. Controller performance on the reboiler.

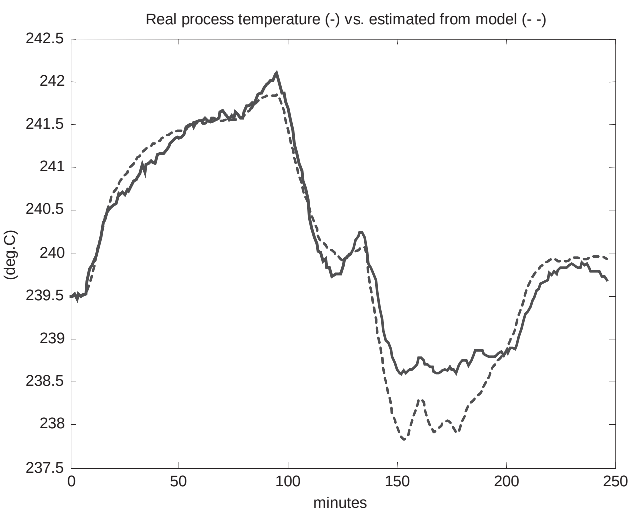

Fig. 15. Validation of the reboiler's DV model.

These will be plotted in future versions of *Sintolab*, since they have proved to be a very good tool when validating the model.

The PID controller was tuned by maximum overshoot of 5% for setpoint change, using the DV model. Fig. 16 shows the controller's performance to setpoint and load disturbance changes. It can be clearly seen that the performance of the new values improves on the old ones significantly. In this case the design criteria has been satisfied, both the quick disturbance rejection and the good setpoint following.

6.2. *Feed drum level control*

This case study deals with one of the most common control problems in any petrochemical plant: material balance control. The feed rate of most process units is affected by the upstream process upsets and operational variations. This feed rate change affects the unit performance negatively. A common way to solve this problem is to place along the process a feed drum, which is responsible for minimizing feed rate changes to the unit by using the surge capacity of the vessel. In order to accomplish this function, a level controller is required.

The tuning of this PID is quite important, since the vessel functionality is fully dependent on the controller performance. Tight control (perfect setpoint following) will transmit all the inlet flow variations to the outlet of the vessel, therefore, making the vessel functionality ineffective. In this case, the average control is required, allowing the vessel level to move between a predefined set of limits, which will minimise the outlet flow variation.

There are special tuning methods (Korchinski, 1995) based on the vessel's geometry. However, it is sometimes difficult to get the inside geometry of the vessel with enough accuracy. This is a key point which will eventually provide an appropriate set of control parameters. For this reason, the testing of the process is a more practical approach of obtaining a model, and then, of applying tuning rules based on the parameters of the model.

Fig. 17 shows the process flow diagram, where a measurement of the inlet flow (FT-IN) is available. The LC controller manipulates the outlet flow valve.

Fig. 18 shows the data used to obtain the dynamic models. It is important to point out that the inlet flow is changing very often, thus affecting the vessel's level; the LC controller mode was set to manual during the test.

The step responses (coefficient models) estimated with *Sintolab* are shown in Fig. 19. The dashed line corresponds to the model (noDV model) that does not consider the load disturbance (inlet flow), whereas the

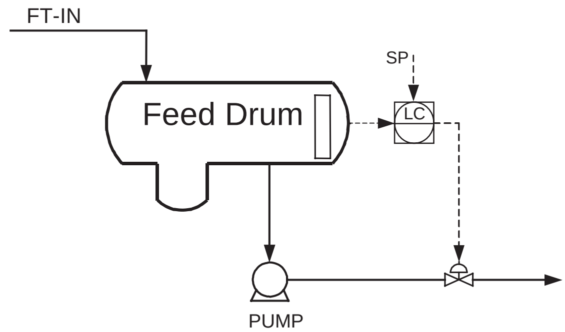

Fig. 17. Feed drum level PID control.

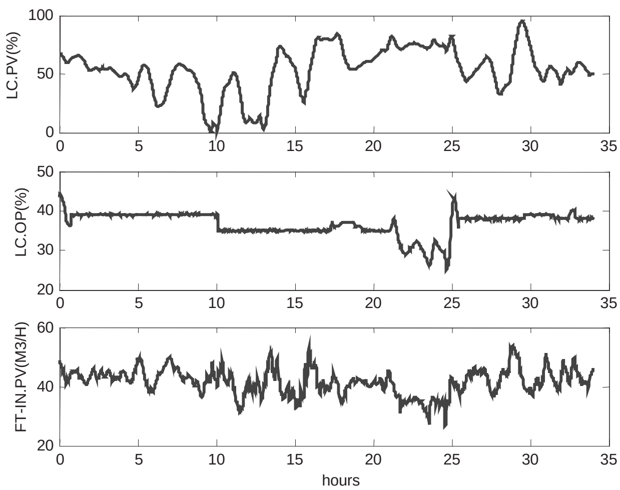

Fig. 18. Test data on the feed drum.

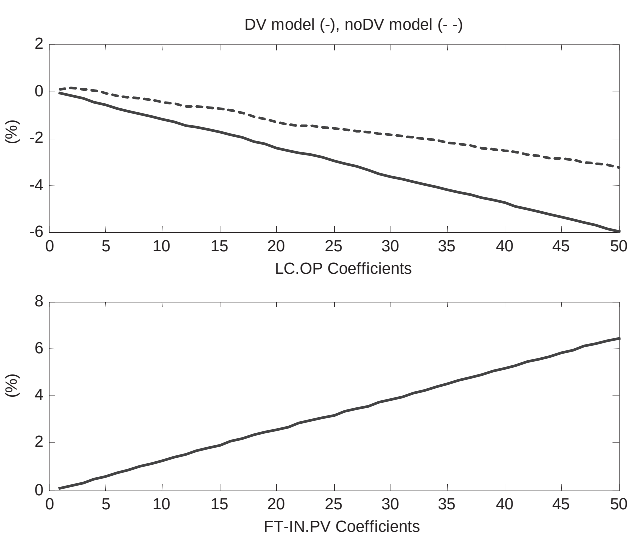

Fig. 19. Dynamic models of the feed drum.

continuous lines represent the model (DV model) that uses the inlet flow as an independent variable. Clearly there is a big difference in the gain of the models that

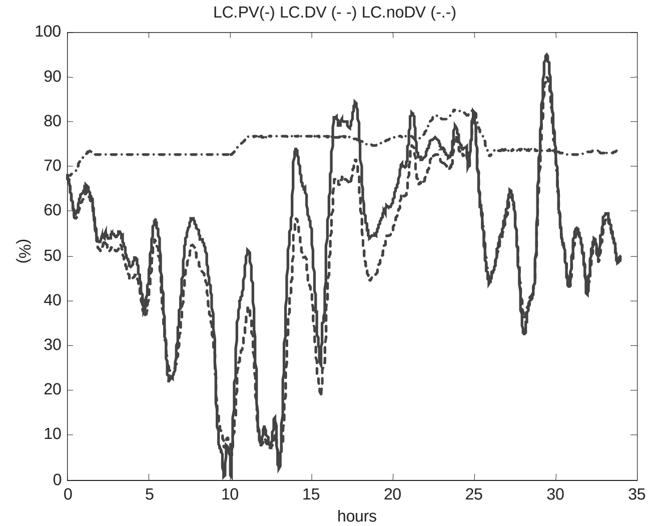

Fig. 20. Validation of feed drum's models.

represent the valve moves (LC.OP) and the vessel level (LC.PV) relationship, see upper diagram of Fig. 19. When considering the inlet flow, the gain is almost double. The question that arises now is: *Which is the right one?*

To answer this question we can do several things. First, we can take a quick look at the step responses of Fig. 19. The slope of the two continuous curves (LC.OP→LC.PV and FT-IN→LC.PV) is almost identical, the sign is different for obvious reasons. This is a good, but not a definite indication of the DV model's validity.

Fig. 20 compares the actual level variations (LC.PV) and the estimates from both models. Certainly, the estimate from the model using the load disturbance (LC.DV) is much better than the one that does not (LC.noDV). In this case, the preferable estimate fits very well, which enables us to foresee a very good control parameters calculation that will satisfy the design criterion.

Fig. 21 shows two closed loop process simulations with the DV model, resulting from a sudden change in both the setpoint and the load disturbance. The design criterion specifies that the level should deviate 10% from its setpoint given a load disturbance of 10 units. The controller tuned using the better model (DV model) satisfies the design criterion, whereas the controller obtained from the other model rejects the load disturbance much faster than required. This is so in the latter case, because the calculated process gain is smaller than the actual one, hence, the controller moves the valve more than it is required.

Finally, it is important to point out that an incorrect tuning of this type of level control loop can make the vessel investment completely worthless. It will clearly degrade the downstream unit control performance.

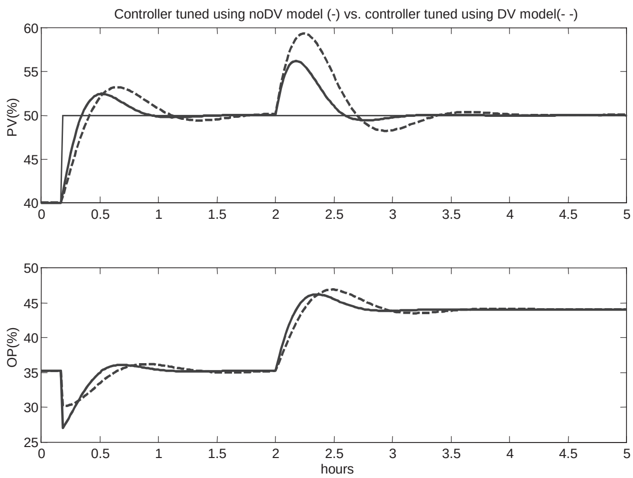

Fig. 21. Comparison in simulation with the DV model of two controller settings for the level control.

7. Conclusions

This paper has described a new tuning tool for industrial PID loops. Its use is almost essential before implementing any MPC controller in order to make sure that all controllers at the lower level are tuned to meet the design criteria.

Using MISO modelling has proven to be a very practical tool when tuning PID loops, even though it is not strictly necessary to obtain perfect models for tuning purposes. It makes the model validation process much easier by requiring a less complex test design, usually of a shorter duration. This last point is particularly appreciated in industrial applications.

Another benefit of modelling the load disturbance is *Sintolab*'s ability to determine the convenience of using a PID algorithm or a more complicated control strategy. This choice would always be more difficult if the load disturbance model is not available.

The seven tuning methods of *Sintolab*, which cover two types of specifications (features of time responses and features of frequency responses), open a lot of possibilities in order to achieve the main goal, which is "to determine the more adequate control parameters exportable to the distributed control system". Even though, only Honeywell's PID algorithms are fully supported by *Sintolab*, it is relatively easy to convert the results for other commercial controllers. At the moment, it is also possible to use Foxboro algorithms.

Sintolab has demonstrated the fruitfulness of a joint venture between the university and the industry. It is the best way to combine industrial practice with a solid theoretical basis. *Sintolab* is being widely used in all the refineries of the group.

Acknowledgements

The current version of Sintolab is the result of seven years of work and research. Many people have been involved in the project throughout its course, none of who should go unacknowledged. Our deepest gratitude to all people from REPSOL, PETRONOR, UNED and UAB who have contributed to the development of *Sintolab*.

Finally, we thank the management of REPSOL-YPF for their support and their permission to publish this paper.

References

Aström, K. J., & Hägglund, T. (1995). *PID controllers: Theory, design, and tuning* (2nd ed.). Research Triangle Park, NC: Instrument Society of America.

Korchinski, W. J. (1995). Advanced level control: Explained and analysed. *Hydrocarbon Processing*. 65–70.

Moore, D., & McCabe, G. (1993). *Introduction to the practice of statistics*. New York: W.H. Freeman and Company.

Morilla, F., & Dormido, S. (2000). Methodologies for the tuning of PID controllers in the frequency domain. In *PID'00 IFAC workshop on digital control: Past, present and future of PID control* (pp. 155–160). Spain: Terrassa.

Morilla, F., & Pastor, R. (2001). *SINTOLAB: Manual de usuario*.

Ogunnaike, B. A., & Ray, W. H. (1994). *Process dynamics, modeling and control*. Oxford: Oxford University Press.

Pastor, R. (2001). *SINTOLAB: Guía de instalación*.

Qin, S. J., & Badgwell, T. A. (1997). An overview of industrial model predictive control technology. In *Fifth international conference on chemical PROCESS control* (pp. 232–256). California: Tahoe.