

## SINTOLAB : THE REPSOL-YPF PID TUNING TOOL

R. Gonzalez-Martin<sup>+</sup>, I. López\*, F. Morilla\*, R. Pastor\*

<sup>+</sup> Dpto de Control Avanzado, PETRONOR, Apdo. 1418, 48080 Bilbao, Spain.  
Phone:34-94-6357185, E-mail: rgonzalezm@repsol-ypf.com

\* Dpto de Informática y Automática, UNED, Avda. Senda del Rey 9, 28040 Madrid,  
Spain. Phone:34-91-3987156, Fax:34-91-3986697,  
E-mail: ilopez@dia.uned.es; fmorilla@dia.uned.es; rpastor@dia.uned.es

**Abstract:** This paper presents a computer tool for tuning PID controllers. What makes this tool different to others available in the market is the ability to model the most important load disturbances. The MISO calculated model enables a better 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, either rejecting disturbances having a fixed setpoint, or following the setpoint, like in Multivariable Predictive Control applications, where the MPC controller is writing its output to the PID controllers setpoint. The tool is the major result of a joint industry-university development. The tool is being heavily used in all the refineries and some petrochemical complex of the REPSOL-YPF group. *Copyright © 2002 IFAC*

**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 placed into the Petrochemical and Refining Industry to support the development of Advanced Control Applications. The main reason has been the maximisation of the production margin or profit, i.e. money. The equation *money makes money* is the main motivation.

It is the Advanced Control Engineer's role to translate the company's overall economics objectives into particular 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 control strategy might be required. This is the case of Multivariable Predictive Controller applications, where usually the higher hierarchy controller is writing its output to a standard PID controller setpoint. This situation is shown in Figure 1.

The question, which the engineer has to deal with, 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.

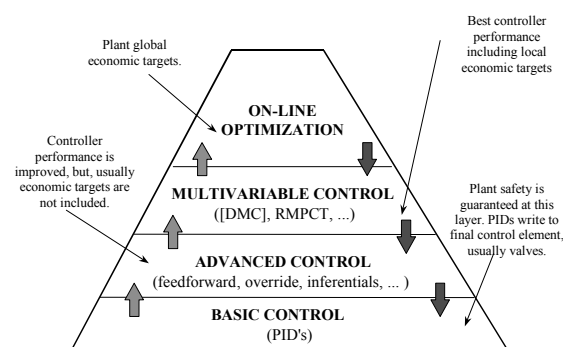


Fig.1. Typical hierarchical control structure

It would be very useful to have a procedure to determine that a more advanced controller is required. This has to be done 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 criteria for that particular loop or not.

Two situations are possible; the best one is where the tuned PID controller is able to do the job properly meeting all the design criteria. The other is the opposite; the PID performance is not good enough, requiring a more advanced controller. Usually those cases will require a MPC controller writing setpoints to a PID. The PID still keeps playing, but its design criteria will be less demanding.

At the beginning of 1994 this was a big concern in REPSOL-PETROLEO and PETRONOR S.A. (companies of the REPSOL-YPF group). The marketplace did not have a product that fulfilled all the requirements. At that time, none of the products 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 evolved during all this time, being now, one of the compulsory tools to be used before any MPC project is implemented.

Section 2 lists the main features included in **Sintolab**. 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. A complete case study is presented in section 6.

## 2. SINTOLAB FEATURES

**Sintolab** is a 32 bits MS-Windows™ application (Morilla and Pastor, 2001; Pastor 2001), which allows tuning and simulation of PID control loops. The basic control structure is shown in Figure 2, where:

**PLANT** represents the dynamical behaviour of the process to be controlled. It includes relationships from the control output signal (OP) and disturbances, to the process variable (PV).

**PID CONTROLLER** represents the element that calculates the control output (OP) based on the setpoint (SP) and the actual process variable (PV). Usually this block reproduces a commercial PID controller algorithm. In REPSOL-YPF the most common PID controller is performed in the Honeywell's TDC-3000™ Distributed Control System.

**Sintolab** includes:

- Tools to manipulate and treat 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 calculate and validate dynamical models of the process based on test data. The model

includes the dynamics of the most important measurable load disturbances.

- Model generation based on first principles for level processes.
- Several tuning methods and different interfaces 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) the study of the open loop response of the system based on predefined inputs b) the study of the closed loop system, it supports setpoint and load disturbances changes c) the evaluation of the controller performance in similar conditions to those present in the actual process.
- Control parameters validation based on the final controller characteristics where the PID is implemented, e.g. limits checking based on type of equation, etc.
- Automatic tuning report generation. This is extremely useful to keep track of all the work done in a particular loop over the time. Knowing the history of the loop is an invaluable information to be used when tuning.

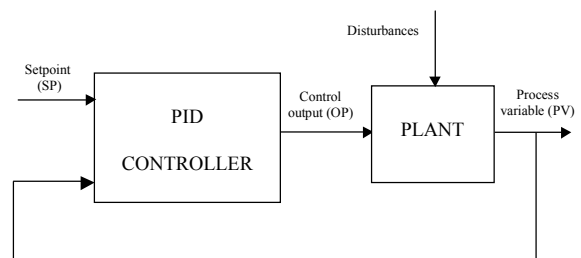


Fig. 2. Basic structure of a PID loop in **Sintolab**.

## 3. PROJECT MANAGEMENT

A project in **Sintolab** is, in its wider sense, a collection of objects used to determine the control parameters of the main object, named **Controller**. Only a main object is included in each project, but a project can have as many basic and auxiliary objects as it is needed. Next, some objects are described and also 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 in the plant. This controller, see Figure 2, reproduces the ideal PID algorithm control, that is considered the standard implementation by ISA (Instrument Society of America). Three equations (PID, PI-D, I-PD), shown in Table 1, are allowed, which are characterised by the following control parameters:  $K_p$  proportional gain,  $T_i$  integral time constant,  $T_d$  derivative time

constant and  $\alpha$  filter factor on the derivative action (fixed to 0.1).

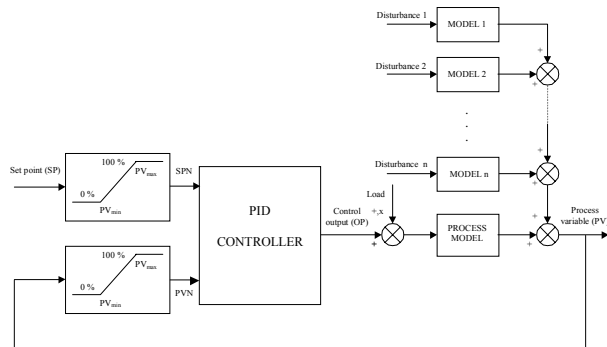


Fig. 3. Control loop model used in *Sintolab*.

Table 1: Equations of the ideal PID control algorithm

Equation	
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	$OP = K_P \frac{1}{T_I s} (SPN - PVN) - K_P \left( 1 + \frac{T_D s}{\alpha T_D s + 1} \right) PVN$

The controller block in *Sintolab* also has other characteristics:

- It calculates and 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 on 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 Figure 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, tuning and simulation.

**Vector.** The vectors are data containers. These data may be plant data from the Distributed Control System (they will be used to estimate models), or any data available in the project, the data generated in the simulations, by example.

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

The plant model, see Figure 3, is composed of a process model block and so many disturbance model blocks as measured disturbances have been recorded. *Sintolab* only contemplates three types of parametric models for the process, presented in Table 2. Level models are also included in *Sintolab* as a special case of the third model, with  $K$  as function of the tank's geometric properties,  $T_1=0$  and  $T_0=0$ .

Table 2: Parametric models

TYPE OF MODEL	Transfer function
First order plus dead time	$\frac{K}{(T_1 s + 1)} e^{-T_0 s}$
Second order plus dead time	$\frac{K}{(T_1 s + 1)(T_2 s + 1)} e^{-T_0 s}$
First order plus dead time and integrator	$\frac{1}{s} \frac{K}{(T_1 s + 1)} e^{-T_0 s}$

**Tuning.** A tuning object represents a possible set of parameters ( $K_P$ ,  $T_I$  and  $T_D$ ) for the controller module. The control parameters calculation in *Sintolab* can be carried out in seven different ways. These methods are summarised in Table 3, joined with its corresponding specifications that are imposed by the user.

**Simulation.** The simulations are experiments performed with a model or a tuning of the project. Each experiment will have its own duration and conditions on the input signals chosen by the user.

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 and 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, helping that the procedure to consume little time.
Robust tuning	Settling time, overshoot and control parameters ranges. Grid size for control parameters are needed too.
Tight control	No specifications are needed. Only are valid for level models.
Average control	The deviation level allowed and the maximum flow disturbance expected. Only are valid for level models.

### 3.3 Main actions.

Each project in Sintolab is built making a series of actions in a predefined sequence. In these actions certain objects are used and others are generated. The main actions (in the correct sequence) are the following ones:

**Controller definition**, which consists of a correct description of the module used in the Distributed Control System, mapping their characteristics into the controller object.

**Data collection**, which represents the collection of data coming from the plant in the PV, OP and disturbances (if there were any) vectors. This action is not necessary for level tuning based on the geometry of a vessel.

**Model estimation and validation**, which consists of generating a representative model plant, using the data collected previously. An exception is done with the level models, where the estimation is based on the tank's geometric properties. In Section 4 more details of the estimation action are given.

**Control parameters calculation**, which consists of determining the PID control parameters that 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 it can also be implicit a conversion of control parameters (Aström and Hägglund, 1995). It takes place when the module in the Distributed Control System implements the interactive PID control algorithm, described by the expression

$$OP = K'_P \left( 1 + \frac{1}{T_I s} \right) \left( 1 + \frac{T_D s}{\alpha' T_D s + 1} \right) (SPN - PVN)$$

## 4. MODELING DETAILS

Whenever the user requests an estimation, he must specify how many coefficients he wants that the step responses have and what type of process model he wants to use. His request causes a total automatic procedure of three steps:

- Check the excitation. If a register of a particular control loop contains all the signals (OP, perturbations and PV) but one of them didn't have variation, *Sintolab* will estimate only parameters for those blocks (process model and disturbances models) where it has found valid experimental data. *Sintolab* does the same when it detects that an input signal of the register have changed, but its changes had have slight influence to the output; This might happen due to the small amplitude of the changes,

or because its frequency content is little significant for the band wide of the process that the signal is exciting.

- Reconstruction of the step responses (models of coefficients). The objective is to estimate the plant step response, having input data (excitation) without any predefined pattern. All the responses will have the same number of coefficients (values) specified by the user.
- Estimation of the parameters of the models. It consists of estimating the parameters of the process and perturbations models that better approximate the reconstructed step responses.

The estimation in *Sintolab* incorporates a final stage, the presentation of the results (textuals, numericals and graphics), to enable the user validating the model. The first validation must be a visual inspection of every step responses, the reconstructed (models of coefficients) and the generated using the parametric models. The second, related to the complete model of the plant, is the reliable validation. It consists of the graphic comparison of the real plant response and the model response with the same experimental data that was used to estimate it or with another set of data. This validation is complemented with a numerical result, the R2 factor. The closer R2 is to one, the better the estimation will be. Additionally, the square error sum between the plant response and the model response is also calculated., a small amount of this figure will represent a good estimate.

Every estimation in *Sintolab* creates log messages as a mean of improving the validation of the model. These messages can be very helpful to determine further actions if the model is not good or does not satisfy the user's expectation.

## 5. TUNING DETAILS

When the user requests a tuning, he must specify what type of controller (PI or PID) prefers, the process model to be used for the calculations, the tuning design criterion, and the corresponding specifications. See the Table 3. This request causes a semiautomatic procedure, since this requires the user intervention.

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

**The solution is not feasible.** It might be possible when tuning by phase or gain margin, usually due to a bad selection of the design frequency or because of the criterion being too exigent. *Sintolab* provides an interface for the selection of the design frequency in order to avoid this situation. It also might appear in the robust tuning tool due to an over specification of the grid size.

**The solution is unique.** This situation is usual in the average control and tight control. Also happens when tuning by phase margin or gain margin if the user specifies the design frequency.

**The solution is the best** that *Sintolab* has reached. It is a specific characteristic of the tuning methods with iterative processes, it happens in the tuning by phase and gain margin and in the tuning by settling time and maximum overshoot.

**The solution is not unique, but one of them is automatically selected.** The user has required *Sintolab* to look for several solutions, but automatically one of them has been selected and the others has been forgotten. It is a specific characteristic of the tuning methods by phase or gain margin, when the user has specified several frequencies inside the set of possible solutions. In this case the control parameters with maximum integral gain are selected automatically.

**There are several solutions,** because the user has required *Sintolab* to look for all of them. The possible solutions are presented to the user in a well-organized way. It is important to keep in mind that a different solution may be chosen later on depending on the user needs. This is a specific characteristic of the robust tuning.

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 reliable validation (although not always possible), is based on the analysis of the temporal and frequency response characteristics. The user can examine the actual achieved specifications and determine the convenience of using that result or not. Among others, *Sintolab* calculates the PM (phase margin), GM (gain margin),  $t_s$  (settling time),  $m_p$  (overshoot ratio).

Only the tuning with iterative processes, that is to say the tuning by phase and gain margin and the tuning by settling time and maximum overshoot, have associated a textual message. The message will show if the quality index, which measures the specifications of relative way, is inside the tolerance or otherwise if the control parameters calculation has been stopped because the algorithm has fallen in a local minimum or has drained the number of iterations.

## 6. A CASE STUDY

This section shows a case study where a difficult PID control is presented. It is a reboiler outlet temperature controller of a distillation tower (TC-1). Figure 4 shows the process flow diagram. The controller manipulates fuel-gas flow to control the temperature. This temperature is manipulated by a MPC controller, so setpoint following is required. But,

there is a big disturbance present in the process; the fuel-gas heat capacity is changing very often, thus, affecting the outlet temperature. AI-1 measures this disturbance. The design criteria also specify quick disturbance rejection.

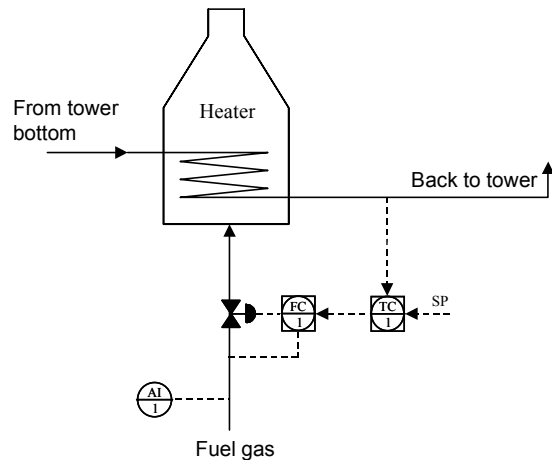


Fig. 4. Reboiler outlet temperature PID control.

Figure 5 shows the test carried out. The loop was set to manual mode at minute 220. A quick step test was applied to the TC-1.OP. The DV curve shows the variation of the fuel gas heat capacity during the test.

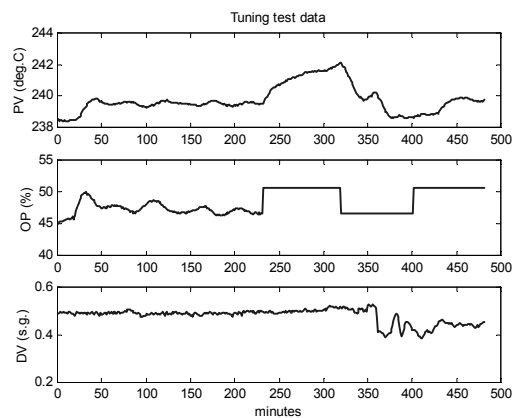


Fig. 5. Test data on the reboiler.

The test data is used now to calculate the dynamical model required for tuning the loop. Figure 6, shows the model estimated temperature vs. the real one. The fidelity of the estimate is acceptable, so the model is validated.

Figure 7 shows the controller performance for setpoint and load disturbance changes. It can be clearly seen that the performance of the new values is much better than the old ones. In this case the design criteria have been satisfied, both the quick disturbance rejection and the setpoint are following. The actual tuning values have been omitted to preserve confidentiality.

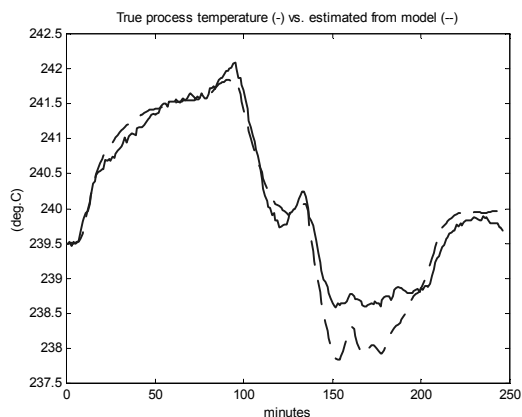


Fig. 6. Validation of the reboiler's model.

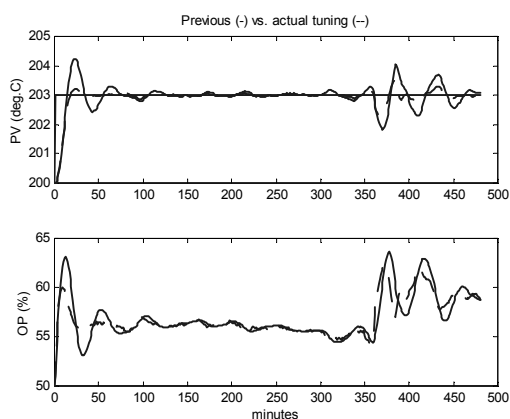


Fig. 7. Controller performance on the reboiler.

## 7. CONCLUSIONS

This paper has shown a new tool for tuning 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 (setpoint following). Sintolab is also very useful in trying to determine the convenience of using a PID algorithm or a more complicated control strategy.

Even though, only Honeywell's PID algorithms are fully supported, it is relatively easy to convert the results for other commercial controllers. Right now, it is also possible to use Foxboro algorithms.

Sintolab has demonstrated that usually the best products come from a combined development between people from university and industry. It is the best way to combine industrial practice with a solid theoretical basis. The tool is being heavily used in all the refineries of the group.

## ACKNOWLEDGEMENTS

The current version of Sintolab is the result of seven years of work. Many people have been involved during all this time, so, the authors would not like to

miss anybody by acknowledging namely. Our deepest gratitude to all people from REPSOL, PETRONOR, UNED and UAB who have contributed to the development of this tool.

Finally, thanks the management of Repsol-YPF for their support and their permission to publish this paper.

## REFERENCES

- Aström, K.J. and T. Hägglund (1995). PID Controllers: Theory, Design, and Tuning (2<sup>nd</sup> Edition). Research Triangle Park, NC: Instrument Society of America.
- Morilla, F. and S. Dormido (2000). Methodologies for the tuning of PID controllers in the frequency domain. PID'00 IFAC Workshop on Digital Control: Past, present and future of PID Control, 155-160, Terrassa, Spain.
- Morilla, F. and R. Pastor (2001). SINTOLAB: Manual de usuario.
- Pastor, R. (2001). SINTOLAB: Guía de instalación.