EBOOK XXL

# The Guide to PID tuning

Accomplish optimal plant performance with PID tuning

IPCOS

Summary

# The Guide to PID tuning

**As a DCS engineer, it's important to understand your PID tuning process. Knowing in advance which steps to take and why to take them is necessary before jumping into the different PID tuning methods available. You'll be able to get the most out of your PID tuning and ensure that you reach your desired plant performance.**

In an industrial plant, tuning PID loops is a time-consuming and difficult job. To achieve optimal results, it's essential to prevent the PID loops to oscillate or overshoot. However, finding the ideal set of parameters for each specific situation is almost an art — it requires experience.

There are many different methods to tune your PID loops; trial and error, Ziegler-Nichols, Cohen-Coon, or model-based PID tuning using software. If you focus on understanding the PID tuning process instead of just using your gut-feeling, you provide more added value to your plant operations.

In this ebook, we will unpack what PID tuning is, how to use it, and how it contributes to optimal plant performance.

Read online version

# Table of contents



IPCOS

# What is PID tuning?

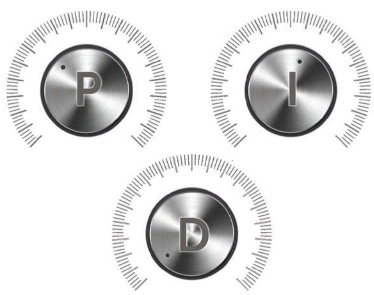
**PID control is a generic control method and workhorse in the automation world: 99% of all automation control loops are PID. Knowing which PID tuning steps to take and why to take them is necessary before jumping into methods and algorithms. As a result, you'll be able to get the most out of your PID tuning and plant performance.**

## 1.1 What does PID tuning mean?

PID tuning refers to the workflow whereby PID parameters are determined based on the PID algorithm used, the open-loop process behavior, and the engineering specifications of the desired closed-loop behavior.

### Optimal PID tuning

Finding the ideal set of P, I and D parameters efficiently is what we call optimal PID tuning. Within the large space of the search parameters, there is only one set that results in optimal performance. Depending on the PID methods you use, the computed parameters can be closer or further away from this optimal set.




### Process behavior

In PID tuning, process behavior is key. Optimal tuning can only be performed when you grab the correct process behavior. Most methods reduce the process to a very simple first-order behavior, although there are many processes that can not be described with these simple models. For example, steam boiler drums, superheaters, processes with long delays, (fed-)batch reactors, and level controllers.

### Engineering specifications

Finally, you shouldn't forget the engineering specifications of the desired process behavior. When setpoint tracking is key, make that your criterion for optimality. When disturbance rejection is key, focus on that. When both are needed, exploit using feedforward. Tune a controller to avoid overshoot if overshoot is not allowed (due to degradation of product quality), or tune a controller to operate in a very broad operating range if that is exactly what happens at the real plant.

In other words, PID tuning means that your control loop has a specific goal which you achieve by using the right P, I, and D parameters. As a result, you'll accomplish the best plant optimization.

## 1.2 Why PID tuning?


PID tuning is necessary in order to have the desired closed-loop control. If, for example, you want to control temperature, then a PID controller needs to be tuned to keep the temperature at the setpoint value.

- The minimum requirement for tuning is that the process can operate in a stable way in closed-loop.
- When you go one step further, PID tuning will control the plant by reducing oscillations. This will result for example, in fewer alarms and fewer operation interventions.
- One step beyond brings you to the point that because of optimal PID tuning the plant efficiency can be boosted, the specific energy consumption can be reduced, more stability can be provided and a minimum amount of alarms and operator interventions.

## 1.3 The key points of PID tuning

PID tuning can result in many benefits. Paybacks of weeks, hours, or even minutes can be achieved when you take the time to adjust the PID parameters to the optimal value. Follow these key steps to tune your PID right the first time:

- Understand the behavior of the process you want to control, for instance by using (multiple) step testing.
- Use the right system ID methods.
- Check your DCS PID algorithm and change it if appropriate
- Make your own engineering wishlist in which you choose between optimal tracking and/or optimal disturbance rejection. Don't forget the robustness and high-frequency gain.
- Compute optimal parameters, implement them, and check the result on the real plant.
- Track and document the changes that you've made.

# PID tuning parameters

Set the right parameters to achieve optimal plant performance

What do these 3 letters mean and how can they influence your plant behavior?

PID is an acronym for Proportional, Integral, and Derivative. These parameters can be used, adjusted, and controlled individually and collectively. You can have, for example, a P controller, a PI controller, or a PID controller. Each parameter can be adjusted and controlled individually and each parameter can be used for specific purposes.

## 2.1 What are PID tuning parameters?

The general PID loop consists of three terms:

1. A proportional action on the error or the PV
2. An integral action on the error
3. A derivative action on the error signal or PV

Picking the right equation and choosing the best parameter value is the essence of achieving optimal tuning.

## 2.2 The P, I, and D parameter

**Description of the Proportional (P) parameter**
The P-action is proportional to the error or the PV. The error (or PV) is multiplied with the proportional gain and added to the controller output. The P-action gives the output a 'kick' in the right direction. If the error value is zero, then the P action is zero. This implies that a controller with only P action needs a non-zero error to have a non-zero output. Accurate tracking is therefore not possible with only P control.

**Description of the Integral (I) parameter**
Consider a plot of the error between PV and SP over time. In mathematics, the "integral" of the error can be interpreted as the surface between the curve and the x-axis and between the y-axis and the current time instant. Every time step, the plot extends a bit to the right. If the error is zero at that time, the surface does not increase and the integral remains constant.

If your error is positive the surface below the error curve will increase, resulting in a higher controller output. The I action will decrease when the error becomes negative.

Typically the I-action will act much slower compared to the proportional action. However, it will bring the error to zero eventually, which the proportional action can't do. So basically, the integral action looks at the past and checks if the error is getting to the setpoint. If not, it's acting on the output. It is going to steer the wheel until you are heading in the intended direction.

**Description of the Derivative (D) parameter**
The integral doesn't have the possibility to predict the behavior of error. The derivative action addresses this problem by anticipating the future behavior of the error.

So, the derivative action is the change of the error. It adds a contribution to the output according to how the error changes. When the error is positive, but is starting to decline the D action, it will reduce the output of the controller. It's the brake that tries to avoid overshoot.

It reduces the oscillations induced by the other two actions. It can speed up the controller to the setpoint that you want to achieve.

It reduces the oscillations induced by the other two actions. It can speed up the controller to the setpoint that you want to achieve. However, the derivative action is not often used in PID tuning. The problem is that it can amplify noise. If the error signal is very noisy, the controller output tends to oscillate a lot. This can negatively affect the lifetime of the equipment like pumps and valves.

# PID tuning methods

Choose the right tuning methods to run a stable production process

How to tune a PID controller quickly and effectively? It's a question that keeps many engineers wondering.

Even though the concept of PID tuning is simple, the mathematics underpinning PID control are complex. Once you have chosen the optimal controller configuration for your problem, the parameters must be tuned. Achieving optimal performance entails selecting the ideal set of numerical values for P, I and D.

In broad terms, there are three approaches to determining the optimal combination of these settings: heuristic tuning, rule-based tuning, and model-based tuning. Each method has its pros and cons. Although many might believe that trial and error methods are free, they often turn out to be extremely expensive.

## 3.1 What is heuristic tuning?

A heuristic tuning method is one where general rules are followed to obtain approximate or qualitative results. The majority of PID loops in the world are tuned with such methods, for better or for worse. The trial-and-error method is an example of heuristic tuning.

**Trial-and-error method**

The trial-and-error method is a relatively easy method, once you get a clear understanding of PID parameters. It steps through the system from proportional to integral to derivative. Usually you start from an existing set of parameters from which you perform small tweaks to improve the response. For new PID loops you start with a rough and safe initial guess.

Basically, one considers:

- The P-action is introduced to increase the speed of the response. Exaggerated P-action results in oscillation.
- The I-action is introduced to obtain a desired steady-state response. The disadvantage is a higher oscillating response over a longer period.
- The D-action is introduced for damping purposes. The disadvantage is the fact that oscillation on a high frequency is more probable, plus the sensitivity to the noise.

How to apply the method depends on tuning a new or existing closed loop.

## Trial-and-error: the pros and cons

Using the trial-and-error method for tuning PID loops has its advantages and disadvantages.

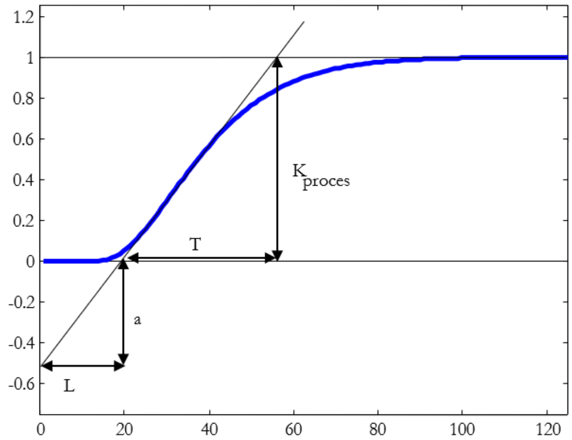
### PRO

- It's a quick and easy way to obtain a reasonable result.
- It has an intuitive approach; no (explicit) assumptions are made about linearity.
- Little knowledge is required. Only simple experiments are required to find suitable controller settings.

### CON

- It's time-consuming. It takes a long time to achieve good performance.
- It doesn't guarantee a (stable) solution for your control problem.
- It makes the system oscillate, which can represent a risk for the entire plant. It can't be applied to all controllers.

# 3.2 What is rule-based tuning?




Rule-based PID tuning methods assume a certain process response to obtain easy mathematical formulas that enable the tuning of a PID controller. The process characteristics can be derived from simple experiments and are used to calculate the PID parameters.

Note that such tuning methods are sensitive to discrepancies with respect to the assumed process response (e.g. first order linear model with delay). Particularly, a big deviation from the assumed process time delay will greatly degrade the actual PID performance. Also, the possibility to define your own control objectives is extremely limited or inexistent.

The PID parameters are presented as the parameters of the general PID control algorithm or the ISA algorithm:

$$u(t) = K_c\left[e(t) + \frac{1}{T_I}\int_0^t e(\theta)\,d\theta + T_D\frac{de(t)}{dt}\right] + u_0$$

| | |
|---|---|
| $u(t)$ | controller output |
| $u_0$ | initial controller output |
| $K_C$ | overall gain |
| $T_I$ | integral time or reset time |
| $T_D$ | derivative time |
| $e(t)$ | error between the setpoint and process value. |

Rule-based tuning applies to several methods: Ziegler-Nichols, Chien, Hrones and Reswick, Cohen-Coon, and Kappa-tau.

## Ziegler-Nichols tuning method
In the industry, Ziegler-Nichols rules are often applied. In many cases, these rules give rise to a control system with acceptable performance trade-offs. However, the design criterion for the Ziegler-Nichols tuning rules is a maximal step overshoot of 25%.

The Ziegler-Nichols tuning method provides two different methods: the step response method and the frequency response method.

| ZN-step response | $K_P$ | $T_I$ | $T_D$ |
|---|---|---|---|
| P-control | $1/a$ | | |
| PI-control | $0.9/a$ | $3L$ | |
| PID-control | $1.2/a$ | $2L$ | $L/2$ |

## Ziegler-Nichols step response method
This method can only be used on stable processes. Open loop tests are required to estimate process characteristics.

## Ziegler-Nichols frequency response method
This method can only be used with a closed loop PID controller. The aim is to push the controller to its stability limits in order to obtain a fast controller which is stable.

Basically, Ziegler-Nichols works well enough when the dead time is small compared to the time constant of the process, and the gain ratio is also small. However, small discrepancies between estimated and actual process characteristics (gain or process delay) can result in an extremely oscillatory or even unstable control loop.

## Ziegler-Nichols method: the pros and cons

Using Ziegler-Nichols to tune PID loops has its advantages and disadvantages.

### PRO

- It's simple, intuitive and you a reasonable performance for simple loops.
- Little knowledge is required. Only a few, fairly simple experiments are required to find suitable controller settings.

### CON

- Quality of tuning is sensitive to non-linearities.
- No handle to define control objectives.
- High proportional gains (due to the 25% overshoot design specification), low integral action with too low damping of the closed loop system and too low robustness against changes in the process dynamics.

### Cohen-Coon tuning method

The Cohen-Coon tuning method is based on the Ziegler-Nichols method, but uses more information from your system in the process. The process is defined by three parameters: the steady state gain a, the time delay L, and the time constant T. This method has significantly better control performance due to the use of more process information.

| Cohen Coon | $K_c$ | $T_I$ | $T_D$ |
|---|---|---|---|
| P | $\frac{1}{a}\left(1+\frac{0.35\tau}{1-\tau}\right)$ | | |
| PI | $\frac{0.9}{a}\left(1+\frac{0.92\tau}{1-\tau}\right)$ | $\frac{3.3-3.0\tau}{1+1.2\tau}L$ | |
| PD | $\frac{1.24}{a}\left(1+\frac{0.13\tau}{1-\tau}\right)$ | | $\frac{0.27-0.36\tau}{1-0.87\tau}L$ |
| PID | $\frac{1.35}{a}\left(1+\frac{0.18\tau}{1-\tau}\right)$ | $\frac{2.5-2.0\tau}{1-0.39\tau}L$ | $\frac{0.37-0.37\tau}{1-0.81\tau}L$ |

$\tau = L/(L+T)$; $a = K_{process}L/T$

## Cohen-Coon method: the pros and cons

Using Cohen-Coon for tuning PID loops has its advantages and disadvantages.

**PRO**

- Avoids errors in estimating a and L by using the gain.
- Works well specifically for systems with a time delay. It gives a quicker closed loop response than the Ziegler-Nichols method.

**CON**

- You can only use the method offline, just like the Ziegler-Nichols first step response.

### Kappa-tau tuning method

The Kappa-tau tuning method is an evolution of the Ziegler-Nichols method. This method is designed to overcome the shortcomings of Ziegler-Nichols, such as high proportional gains and the rules providing poor results for systems with long normalized dead time.

| $f(\tau)$ | Slow controller | | | Fast controller | | |
|---|---|---|---|---|---|---|
| $\downarrow$ | $a_0$ | $a_1$ | $a_2$ | $a_0$ | $a_1$ | $a_2$ |
| $aK_c$ | 0.29 | -2.7 | 3.7 | 0.78 | -4.1 | 5.7 |
| $T_i/L$ | 8.9 | -6.6 | 3.0 | 8.9 | -6.6 | 3.0 |
| $\alpha$ | 0.81 | 0.73 | 1.9 | 0.44 | 0.78 | -0.45 |

$$f(\tau) = a_0 \exp(a_1 \tau + a_2 \tau^2)$$

## Kappa-tau method: the pros and cons

Using Kappa-tau to tune PID loops has its advantages and disadvantages.

**PRO**

- Less oscillatory response.
- Designed for load disturbance response. You can deal with setpoint and disturbance response; it offers the opportunity to distinguish between the two of them.
- Results in optimal disturbance rejection with no overshoot.
- The tuning parameter for the design is the sensitivity of the controller towards process disturbances.

**CON**

- It can’t define control objectives or closed loop performance requirements due to 0% overshoot.

### Lambda tuning method

The Lambda tuning method owes its name to the Greek letter lambda (λ). It dictates how long the controller is allowed to spend on the settling time. The Lambda tuning method assumes the same kind of response as the Ziegler-Nichols method.

| Lambda | $K_c$ | $T_I$ |
|---|---|---|
| PI | $\frac{T}{K_p(\lambda + L)}$ | $T$ |

Typically $\lambda = 3 \cdot \max(L,T)$ (very stable loop)

Reduce $\lambda$ to get a faster response

## Lambda tuning method: the pros and cons

Using Lambda for tuning PID loops has its advantages and disadvantages.

**PRO**

- Enables choosing a desired closed-loop time constant, i.e. how fast the controller responds.
- Works well specifically for systems with a large time delay (dead time is close to the process time constant).
- Results in high robustness against changes in the process dynamics, including non-linearities.
- Results in a response with no overshoot.

**CON**

- Results in a slow rejection of disturbances, especially for slow systems.
- It can't define control objectives and is limited in closed loop performance requirements.
- Only suited for PI controller tuning. The parameter derivative cannot be taken into consideration.

## 3.3 What is model-based tuning?

Model-based tuning or optimization-based PID tuning allows you to obtain your P, I, and D parameters optimally using a model of your system. Engineering specifications of the closed loop behavior are taken into account.

Model-based PID tuning is the method that allows you to work according to a structured tuning process that considers both your process behavior and your control needs. Heuristic and rule-based tuning require an iterative process. Methods such as the Ziegler-Nichols give reasonable results in many (simple) cases, but aren't able to provide the same structured process and production results as model-based PID tuning.

The model-based tuning method may seem more time-consuming, but once you have set the right parameters for your PID loops you'll see the benefits immediately. And these benefits will remain for a long time. After setting your PID controller right the first time, you don't have to look at it again unless something changes in the process.

### Model-based method: the pros and cons

Using the model-based method for tuning PID loops has its advantages and disadvantages.

**PRO**

- Allows a structured tuning process that considers both your process behavior and your control needs.
- Enables a balance between the engineering objectives' performance and robustness.
- It's a flexible method. It will search for the optimal solution close to your requirements. You can compare and test scenarios.

**CON**

- You need to follow a strict workflow. The method requires making explicit how you want the process to behave in closed loop.
- Requires you to identify a sufficiently accurate model, otherwise you will never get the right loop tuning.

# Comparing different methods: a PID tuning example

Discover the limitations and benefits of heuristic, rule-based, and model-based PID tuning

Quickly reacting to process oscillations is very important for a DCS engineer. It's crucial to choose the right PID tuning approach and solve the problem.

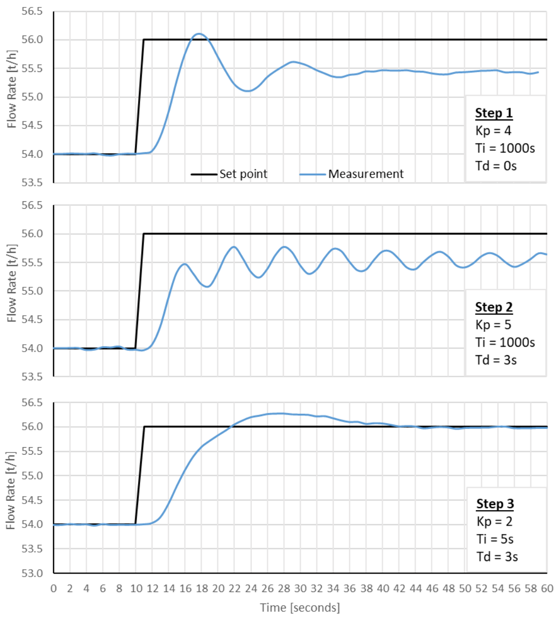
Here you will learn how to use three different approaches in practice:

- **Heuristic tuning**
  Learn about the trial-and-error method and why it often fails to deliver satisfactory results.
- **Rule-based tuning**
  Learn about the most widely used method: Ziegler-Nichols. Why this method provides sometimes good results, but unfortunately also poor results in some cases.
- **Model-based tuning**
  Discover the model-based approach by using PID tuning software. This approach closes the gap between your control needs and the set of values.

## 4.1 Example of heuristic tuning

You've probably dealt with critical PID controllers that cause unwanted process oscillations and frequent alarms. Imagine you came across such a situation with a simple control loop. For example: the flow in a pipe is controlled by adjusting the opening of the valve. And for this specific flow, the operators have requested a fast response controller with no overshoot.

You decide to use the trial-and-error approach: an iterative method for PID tuning. You obtain the three parameters from a series of steps on the controller setpoint while assessing the controller performance. Simple, right?




## Heuristic tuning step-by-step

After you check the controller equation in the DCS (a generic non-interacting PID equation in seconds), you need to perform the following steps:

- Step the setpoint a couple of times with the loop in auto mode and only a small proportional action (Kp). Double this action with every consecutive step until the oscillation is just acceptable for the final closed loop.
- Double the derivative action (Td) with every new step to add damping. Increase Kp and Td until the performance presents faster oscillations.
- Finally, halve the integral action (Ti) in every successive step. When the response becomes too oscillatory, decrease Kp to remove the oscillations.

The figure above shows the controller performance after each point. In the end, you achieved the initial control objectives. But, as you can imagine, the panel operator wasn't thrilled about the oscillations created in the process. Moreover, the iterations were very time-consuming.

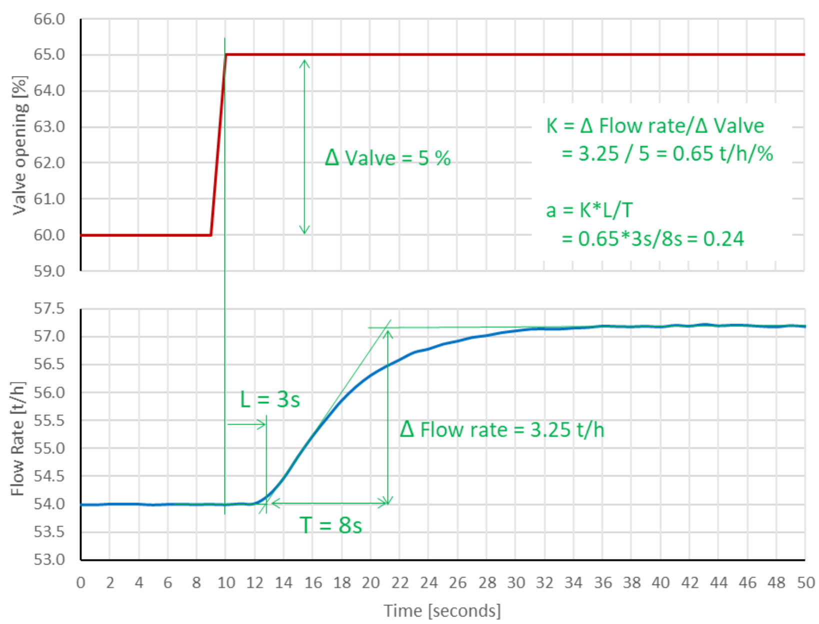
# 4.2 Example of rule-based tuning

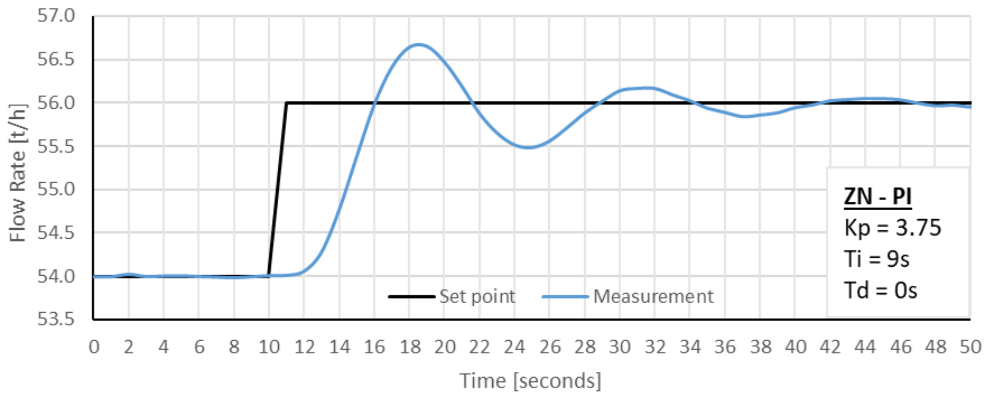
Ziegler-Nichols (ZN) is one of the most widely used ruled-based PID tuning methods. However, it can result in poor controller performance when misused. Reckoning its limits and possibilities can be very useful for a DCS engineer like you!

ZN rules were designed for fast rejection of disturbances with a "quarter amplitude damping" control objective. In other words, the tuning tries to reduce to a quarter the amplitude of the error between setpoint and measurement with each successive cycle. Additionally, the ZN rule also assumes a PID interacting algorithm (or a PI non interacting) and a first order model of the process with a small time delay.
Let's take again the example of a flow controller. With the loop in manual, a step change in the valve opening provided the response characteristics shown in the figure to the right. Once approximated the response to a first order model with delay, the controller parameters were calculated using the following ZN tuning rules:

- P controller: Kp = 1/a
- PI controller: Kp = 0.9/a, Ti = 3*L
- PID controller: Kp = 1.2/a, Ti = 2*L, Td = 0.5*L

where a = K*L/T with K the process gain, L the dead time and T the time constant.






The above and on previous page shows the resulting PI controller (Kp = 3.75, Ti = 9s) tracking a setpoint change. Panel operators don't want any overshoot, but you can't translate that into the parameters with the ZN rules. Controller performance is typically oscillatory and presents very narrow robustness margins. Specifically, it's sensitive to larger gains or larger process time delays (process time delay is greater than half the process time constant). This is the main reason why the resulting tuning is not likely to withstand in an industrial environment.

Other tuning rules have been developed to overcome some of the limitations of ZN such as Cohen-Coon (larger process delays) or Lambda tuning (very stable and robust performance). However, recent model-based PID tuning software allows for a faster and robust tuning based on your control needs.

## 4.3 Example of model-based tuning

You want to quickly transfer your control needs — the desired performance — into a set of values in the PID controller. While this task is limited and cumbersome with other tuning methods, model-based PID tuning excels at closing that gap.

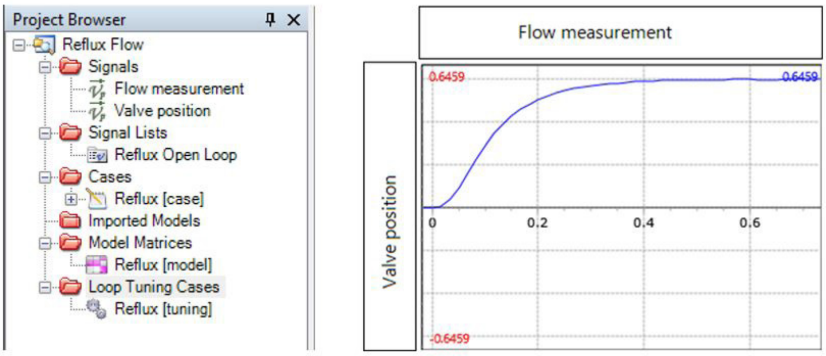
In a systematic approach, a model of the process is identified from open-loop data to be subsequently used for simulation and optimization of the controller performance subject to equipment and/or process constraints. Again, we consider the previous example of a critical flow controller.

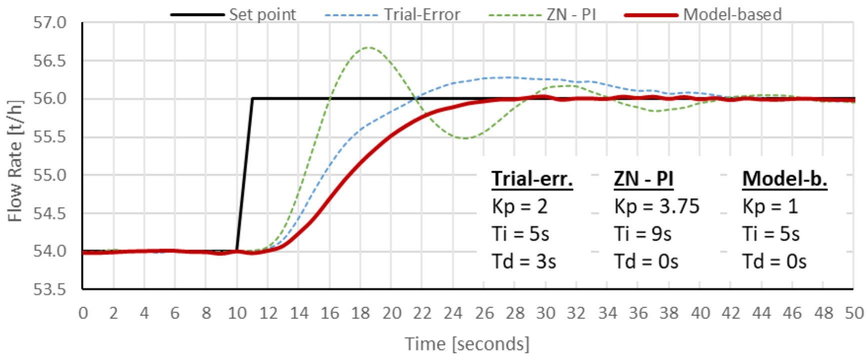
**Model-based tuning step-by-step**

The first step is to place the loop in manual mode and perform a few step changes in the valve opening to collect the flow rate response. Then you use PID tuning software to identify a model of the process (case and model matrix). The picture below shows the typical workflow in model-based tuning and the estimated high order model with the same process gain found in the Ziegler-Nichols rule section (K = 0.65 t/h/%).

Next, you need to create a loop tuning case containing all the relevant characteristics of the loop described in the previous sections. For instance, the process dynamics (process model), the DCS algorithm (generic non-interacting PID equation in seconds), flow and valve ranges, control objectives (setpoint tracking at maximum speed with no overshoot), and robustness (gain and deadtime margins). Then, you need to run the optimization and check the simulated results.

# 4.4 Why use software for PID tuning?




The figure shows the setpoint tracking performance of the optimal PI controller (Kp = 1, Ti = 5s) compared with the trial-and-error method and the Ziegler-Nichols step response method. Clearly, it's the only controller without overshoot and the one that creates the least repercussions on the running process.

All in all, panel operators will be delighted with the new tuning and the loop runs efficiently. Model-based PID tuning with the help of PID tuning software might be the best solution to run a smoothly operating plant.

# PID tuning software

Achieve efficiency, structure, and overview to head straight for your goal

You are never truly done tuning your PID loops. You can go a long way tuning PID controllers just based on your gut feeling and experience. But if you want to discover the behavior of the loop and bring structure to your tuning process, INCATools (PID tuning software) offers you the right solution.

## 5.1 Model-based PID tuning

Model-based PID tuning or optimization-based PID tuning follows a predefined workflow. You first need to decide which type of controller you want to obtain. Then you retrieve data acquired from the system. Next, you identify a model that describes the reaction of the outputs to the inputs. Finally, the software determines  the tuning parameters that meet your objectives the best.

Model-based PID tuning is the only way that naturally leads to this structured and first-time-right tuning process. Heuristic and rule-based tuning will always remain iterative processes. Methods such as Ziegler-Nichols and Cohen-Coon give reasonable results in many (simple) cases, but will never be as structured as model-based tuning. When dealing with more complex loops none of the other methods actually generate satisfactory results.

Identifying models for every loop might feel time-consuming. It may seem faster to adjust the P, I, and D manually, but you'll benefit in the long run. You'll only have to observe your PID loop once instead of tweaking it every single time. Model-based PID tuning is efficient because the tuning process is structured, allowing you to detect and solve issues much faster.

## 5.2 Model-based tuning: the perfect fit for PID tuning software

PID tuning is an art which requires experience. Even if you succeed in identifying all process models, it's impossible to manually set them right the first time. PID tuning software adjusts to your specific DCS and loop type, so that you don't have to depend on your own knowledge of the exact equations used by the loop.

The biggest challenge of tuning your PID controllers is choosing the balance between two important engineering objectives: performance and robustness. With PID tuning software you can specify how to make that consideration and obtain the perfect PID loop.

## 5.3 INCATools: state-of-the-art PID tuning software

inca tools

INCATools is a set of process control tools that optimize your plant operations. It's designed to make your job easier by ensuring you can overcome your challenges, even if you don't have the academic background or the skillset of an experienced DCS engineer. On top of overcoming more basic tuning challenges with individual PID loops, you can also tune interacting loops and use cascade control tuning.

## 5.4 The benefits of INCATools

Both INCATools' packages, PID Tuner and AptiTune have many benefits. They help you calculate the right parameters, simplifies the process of control improvement of individual PID loops, and helps you accomplish the best results faster.

**Benefits:**

- Reproducibility of your actions, allowing you to make a translation between objectives and parameters.
- A more structured and stable production process and more efficient plant operations.
- No experience is needed, the tool will guide you through your desired process
- Fewer process oscillations resulting in less wear and tear on equipment.
- Fewer alarms and operator interventions.
- Increased product quality.

# Learn more about PID tuning software, how it stabilizes your plant and reduces alarms and operator interventions

Walking through the steps of tuning a PID controller with a particular approach can offer you perspective. Request a 30-minute live demo and gain insights into what you can achieve with INCATools PID Tuning. It's all you need to get your PID tuning first time right.

With INCATools PID Tuning you'll be able to ...

- Reduce process oscillations resulting in less wear and tear of the equipment
- Reduce alarms and operator interventions
- Achieve more efficient plant operations
- Put all your PID's in auto mode

It's all you need to get your PID tuning first time right. Request your free, no-obligation demo and get an online demo that's tailored to your specific needs. Get access to your 30-min live demo today.

Request live demo

# About IPCOS

IPCOS is a services provider that maximizes the performance of its customers' assets in the international energy and chemical industries. IPCOS experts implement tailor-made, state-of-the-art technology to increase plant operations efficiency in downstream markets, and integrate information technologies into customer workflows to improve oilfield production and reservoir management in upstream applications. With an extensive track record spanning more than 500 projects in 50 locations in 30 countries, IPCOS is trusted to deliver unique engineering-based, technology-independent solutions that combine domain knowledge with technology know-how.

IPCOS Plant Operations Excellence (POE) provides consulting, technology integration, project delivery, and application maintenance services to optimize customers' plant operations, using technology solutions including advanced process control, alarm management and process modelling. On an international scale, IPCOS POE services the international refining, chemicals and gas processing industry.
IPCOS POE combines deep knowledge of processes and the related operation, unique understanding of software technology, and strong domain experience to design and integrate tailor-made practical solutions that drive plant operations efficiency.

Are you looking for ways to achieve safer operations, increased quality control and stable plant performance? We are happy to advise you about how control and the right tooling can help you accomplish optimized plant performance.

Talk to a control expert

IPCOS