

## Interactive Learning Modules for PID Control

### Using Interactive Graphics to Learn PID Control and Develop Intuition

JOSÉ LUIS GUZMAN, KARL JOHAN ÅSTRÖM, SEBASTIAN DORMIDO, TORE HÄGGLUND, MANUEL BERENGUEL, and YVES PIGUET

Interactive tools can be used to complement books and lectures [1]–[4]. This article describes three interactive learning modules that are designed to develop intuition as well as a working knowledge of proportional-integral-derivative (PID) control. These three modules comprise a package called interactive learning modules for PID (ILM-PID). By illustrating concepts such as tuning, robustness, loop shaping, and antiwindup, ILM-PID can be used for demonstrations, exercises, and self-study.

The main objective of the interactive modules is to explain basic concepts of PID control without considering implementation aspects. Although most PID controllers are implemented as sampled-data control systems, analysis and design are traditionally performed in continuous time assuming that the sampling rate for subsequent digital implementation is sufficiently fast. Implementation issues, such as aliasing, selection of the sampling time, signal prefiltering, influence of the discretization algorithms, and bumpless parameter changes, may be the aim of a future interactive modules focused on implementation aspects for PID control.

The modules of ILM-PID have menus for selecting process transfer functions and controller structures. In addition, parameters can be set, and results can be stored and loaded. A graphic display of time and frequency responses is a central part. The plots can be manipulated directly by dragging points and lines and by using sliders. Parameters that characterize performance and robustness are displayed. Each module has two icons called *Instructions* and *Theory*. *Instructions* provides access to a document that contains suggestions for exercises, while *Theory* provides access to relevant theory by means of the Internet. The modules are implemented in Sysquake [5], a Matlab-like language with fast execution and capabilities for interactive graphics.

The following sections describe three modules that illustrate closed-loop fundamentals (PID Basics), loop-shaping design (PID Loop Shaping), and integrator windup (PID Windup). Readers are encouraged to visit the Web site [6] to experience the interactive features of ILM-PID. The modules are available for Windows, Mac,

and Linux operating systems and can be freely downloaded from the Sysquake Web site [7] as described in “Downloading and Using ILM-PID.”

### PID BASICS

The module PID Basics is designed to explore the properties of a simple feedback loop by showing the time and frequency responses of a closed-loop system and demonstrating how these responses are influenced by the choice of controller parameters.

A block diagram of a basic feedback loop is shown in Figure 1, where  $P$  and  $C$  are the process and controller transfer functions, respectively, and  $F$  is the filter transfer function for the setpoint. The system has three inputs representing the setpoint  $y_{sp}$ , the load disturbance  $d$ , and the measurement noise  $n$ . It is assumed that the load disturbance acts at the process input and that the measurement noise acts at the process output. The controller must reduce the effect of the load disturbance and make the process variable  $x$  follow the setpoint  $y_{sp}$ , while not injecting too much measurement noise. In addition, the closed-loop system must be insensitive to variations in the process dynamics.

At least three signals are of interest, namely, the process output signal  $x$ , the measured output signal  $y$ , and the control signal  $u$ . Tracing signals in the block diagram in Figure 1 gives the relations

$$X = \frac{PCF}{1+PC}Y_{sp} + \frac{P}{1+PC}D - \frac{PC}{1+PC}N$$

$$= FTY_{sp} + PSD - TN, \quad (1)$$

$$Y = \frac{PCF}{1+PC}Y_{sp} + \frac{P}{1+PC}D + \frac{1}{1+PC}N$$

$$= PCFY_{sp} + PSD + SN, \quad (2)$$

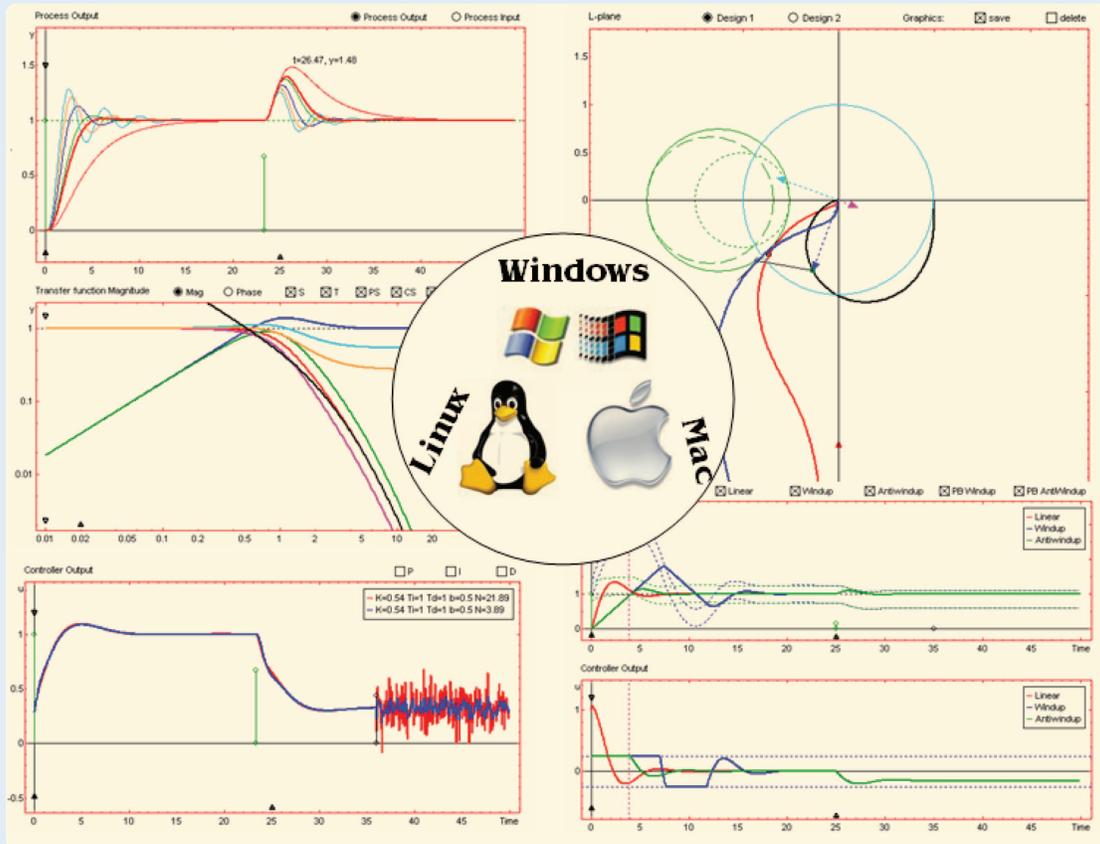
$$U = \frac{CF}{1+PC}Y_{sp} - \frac{PC}{1+PC}D - \frac{C}{1+PC}N$$

$$= CFSY_{sp} - TD - CSN, \quad (3)$$

where capital letters denote Laplace transforms of the corresponding time functions,  $S = 1/(1+PC)$  is the sensitivity function, and  $T = PC/(1+PC)$  is the complementary sensitivity function. Notice that the input-output relations are completely characterized by the six distinct transfer

## Downloading and Using ILM-PID

Interactivity, which is the main feature of the tools described in this work, is difficult to explain in written text. The best way to appreciate the tools is to use them. We strongly recommend that the reader download and use them in parallel with reading this article. Executable versions for PC, Mac, and Linux are freely available for download from the Calerga Web site [7]. No licenses are required, and the executable modules can be freely distributed to students and colleagues.



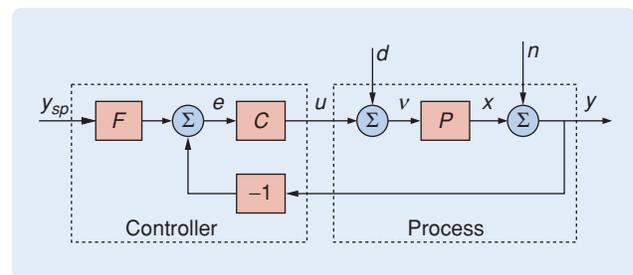
functions in (1)–(3). These transfer functions are called the gang of six in [8]. To analyze the closed-loop system it is necessary to consider all six transfer functions.

The time responses of the six transfer functions are illustrated by showing the response of the process output and control signals to a step in the setpoint, a step in the load disturbance, and wideband measurement noise, as illustrated in Figure 2. A mix of time and frequency responses can also be displayed.

Process models in the form of rational transfer functions with a time delay can be chosen from a menu that provides a collection of transfer functions. An arbitrary transfer function can also be entered using the standard Matlab format. The process gain and time delay can be changed interactively using sliders. The PID controller has the structure

$$U = K \left( bY_{sp} - Y + \frac{1}{sT_i} (Y_{sp} - Y) - \frac{sT_d}{1 + sT_d/N_d} Y \right),$$

where  $K$  is the proportional gain,  $T_i$  is the integral time,  $T_d$  is the derivative time,  $N_d$  is a parameter of the derivative term, and  $b$  is the setpoint weight.



**FIGURE 1** Basic feedback loop having two degrees of freedom.  $P$  and  $C$  are the process and controller transfer functions, respectively, and  $F$  is the filter transfer function on the setpoint. The variable  $y_{sp}$  is the setpoint,  $e$  is the tracking error,  $u$  is the controller output,  $d$  is the load disturbance,  $x$  is the process variable,  $n$  is the measurement noise,  $y$  is the measured output signal, and  $v$  is the controller output corrupted by the load disturbance  $d$ .

## The Interactive Tool

The main screen of the tool is shown in Figure 3. The process is characterized by the parameter group located on the left-hand side of the screen, just below the icons (see Figure 3). The process is shown symbolically together with several interactive elements for changing the representative parameters of the process. The transfer function in Figure 3 is

$$G(s) = \frac{K_p}{(s + 1)^n},$$

where the gain  $K_p$  and order  $n$  are the interactive elements, with numerical values  $K_p = 1$  and  $n = 4$ .

When the user modifies any plant parameter, the symbolic representation of the process transfer function is immediately updated, and its effect is reflected on the remaining graphic elements.

Five buttons are available for selecting the desired controller. The buttons correspond to proportional (P), integral (I), proportional-integral (PI), proportional-derivative (PD), and proportional-integral-derivative (PID). Several sliders are available below the radio buttons for modifying the controller parameters. The number of sliders shown depends on the chosen controller. For instance, Figure 3 shows five sliders since the PID controller is selected.

## Performance and Robustness Information

Parameters that characterize performance and robustness are also displayed on the screen. The performance criteria are based on the setpoint response, the load disturbance response, and the noise response. The setpoint response is characterized by the integral absolute error (IAE) and the

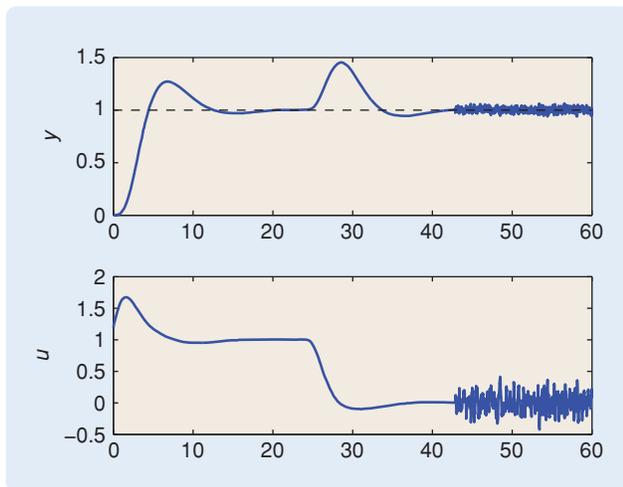
overshoot (overshoot). The load disturbance response is characterized by the integral absolute error (IAE), the integral gain  $k_i = K/T_i$  (ki), the maximal error (emax), and the time to reach the maximum (tmax). The integral absolute errors and the maximal error values are normalized to unit step changes in setpoint and load disturbances. The response to measurement noise is characterized by the standard deviations of the process variable  $x$  (sigma\_x), measured output  $y$  (sigma\_y), and control signal  $u$  (sigma\_u). The robustness measures are maximal sensitivity (Ms), maximal complementary sensitivity (Mt), gain margin (Gm), and phase margin (Pm). This information can be duplicated to compare two designs, as shown below. A more detailed description of these measures can be found in [8].

## Graphics

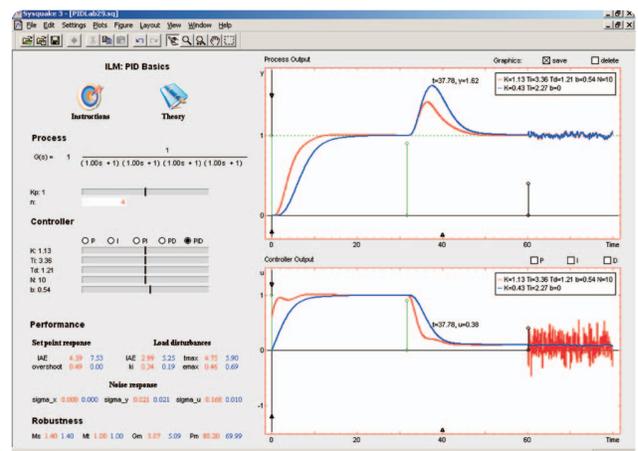
Two graphics are shown on the right-hand side of the tool (Figure 3). Three representation modes can be selected from the Settings menu. These modes are time domain, frequency domain, and frequency/time domain.

The time domain mode is shown in Figure 3, where the time responses for the system output (Process Output) and input (Controller Output) are displayed. The initial part of the plots ( $0 < t < 30$ ) shows the response to a step change in the setpoint represented by the transfer functions  $FT$  and  $CFS$  in (1)–(3). The middle portions of the plots ( $30 < t < 60$ ) show the response to a step in the load disturbance represented by the transfer functions  $PS$  and  $T$  in (1)–(3). The last portions of the plots ( $t > 60$ ) show the response to wideband measurement noise, which is represented by the transfer functions  $S$  and  $CS$  in (1)–(3).

Several elements on the graphics are available for interacting with the application. The vertical green line at time



**FIGURE 2** Control system responses illustrating basic feedback system properties. To analyze the feedback loop, it is essential to consider six responses. These responses, which are referred to as the gang of six [8], are described by transfer functions in (1)–(3). One way to present this information is to show the process output  $y$  and the controller output  $u$  for step commands in setpoint and load disturbances, as well as the response to sensor noise, as shown here.



**FIGURE 3** The user interface of the module PID Basics. The plots show the time response of the transfer functions in (1)–(3) [8]. Several graphical elements, shown on the same screen, are used to interactively analyze feedback fundamentals using PID control. This example provides a comparison between PI (blue) and PID (red) controllers.

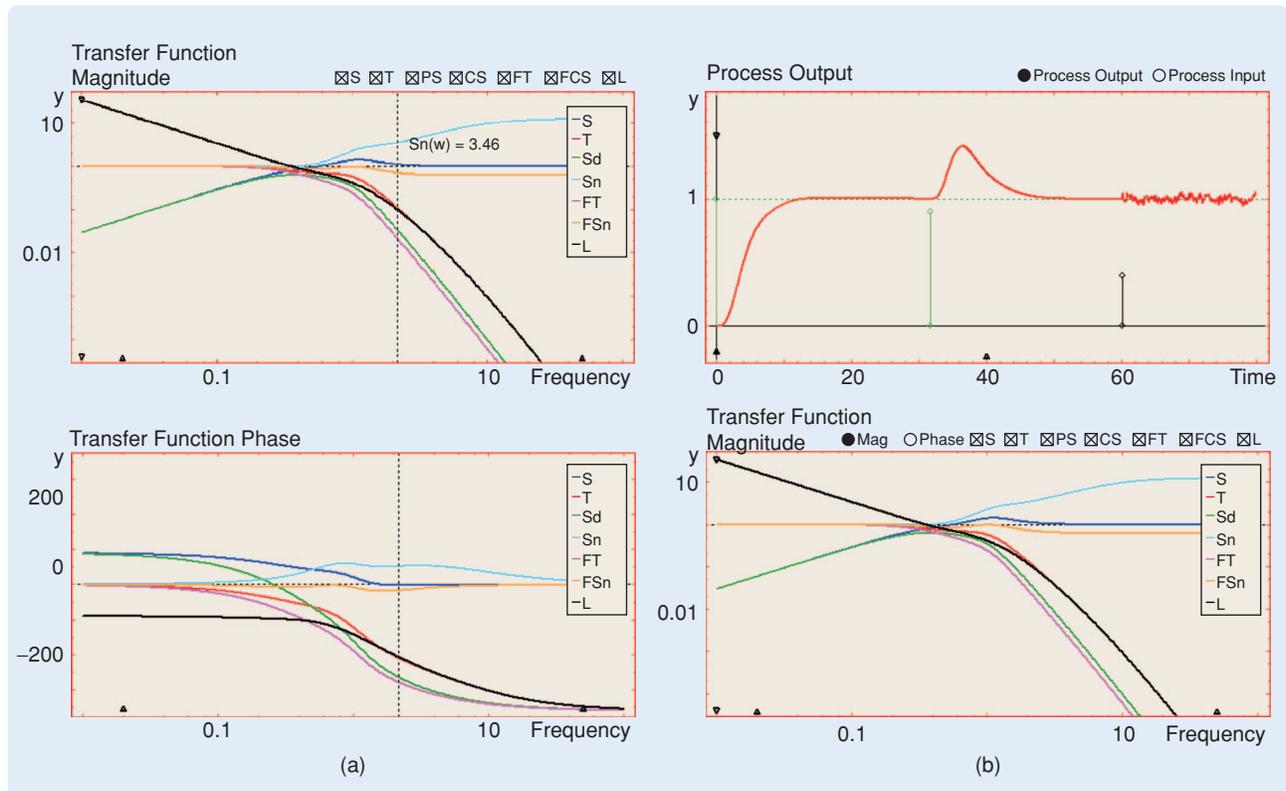
$t = 0$  allows the setpoint amplitude to be modified. The green and black vertical lines located in the middle of the graphics allow setting the value and time instant for load disturbances and measurement noise, respectively. The vertical and horizontal scales can be changed using the black triangles ( $\blacktriangle$ ,  $\blacktriangledown$ ) available in the graphics. For instance, in Figure 3, the setpoint is set to one, the load disturbance is set to 0.9 at  $t = 32$ , and the measurement noise is set to 0.02 at  $t = 60$ . It is also possible to find the value for the input or output signal at a specific time by placing the mouse over the curve. Figure 3 shows an example in which, at the time instant  $t = 37.78$ , the output and input signals are 1.62 and 0.38, respectively. All of these options are available in both graphics, that is, Process Output and Controller Output.

The checkboxes save and delete above the Process Output graphic provide the ability to store a simulation for comparison. When the save button is selected, the current design is frozen and displayed in blue, and a new design in red appears, allowing the two designs to be compared. Performance and robustness parameters are duplicated, displaying the values in blue and red colors associated with each design. The Process Output and Controller Output graphics indicate the values of the controller parameters for both designs. Figure 3 presents an example that

compares the response of PI ( $K = 0.43$ ,  $T_i = 2.27$ ,  $b = 0$ ) and PID ( $K = 1.13$ ,  $T_i = 3.36$ ,  $T_d = 1.21$ ,  $b = 0.54$ ,  $N_d = 10$ ) controllers. Although the PID controller provides a better response to load disturbances by reacting faster, the noise also generates more control action. The delete option can be selected to remove a design. If the transfer function of the process or an input signal such as a setpoint, load disturbance, and measurement noise are altered, both sets of results are affected simultaneously. Only two designs are stored to keep the user interface simple.

Additional options for the time-domain mode are shown above the Controller Output graphic. These options show the proportional (P), integral (I), and derivative (D) signals of the controller.

The frequency domain mode is shown in Figure 4. When this mode is selected from the Settings menu, the left side of the tool remains unchanged. However, in this case the time responses are replaced by the magnitude and phase plots Transfer Function Magnitude and Transfer Function Phase. The vertical and horizontal scales can be interactively modified in the same way as in the time domain. The magnitude and phase for a specific frequency can be found by placing the mouse over the signals as shown in Figure 4(a).



**FIGURE 4** Time- and frequency-domain analysis using the interactive tool. (a) Frequency domain. The graphical part of PID Basics is shown for the frequency-domain mode, where the Transfer Function Magnitude and the Transfer Function Phase graphics are displayed. In this mode the user can study the transfer functions in (1)–(3) in the frequency domain using checkboxes placed above the Transfer Function Magnitude graphic. (b) Time and frequency responses, simultaneously. Above the graphics, the two buttons let the user choose between the output or input for the time domain, and magnitude or phase for the frequency domain.

The frequency response for the gang of six transfer functions and the open-loop transfer function  $L(i\omega) = P(i\omega)C(i\omega)$  can be shown in the graphics using checkboxes placed above the Transfer Function Magnitude graphic. In Figure 4(a), all transfer functions are displayed.

Time and frequency responses can be shown simultaneously, as illustrated in Figure 4(b). The upper part represents the time responses, while the lower part shows the frequency responses. The default screen shows the output and the magnitude for the time and frequency domains, respectively. Above the graphics, the two buttons let the user choose between the output or input for the time domain and magnitude or phase for the frequency domain. This mode is useful since it is possible to view the effect of parameter modifications on both domains simultaneously.

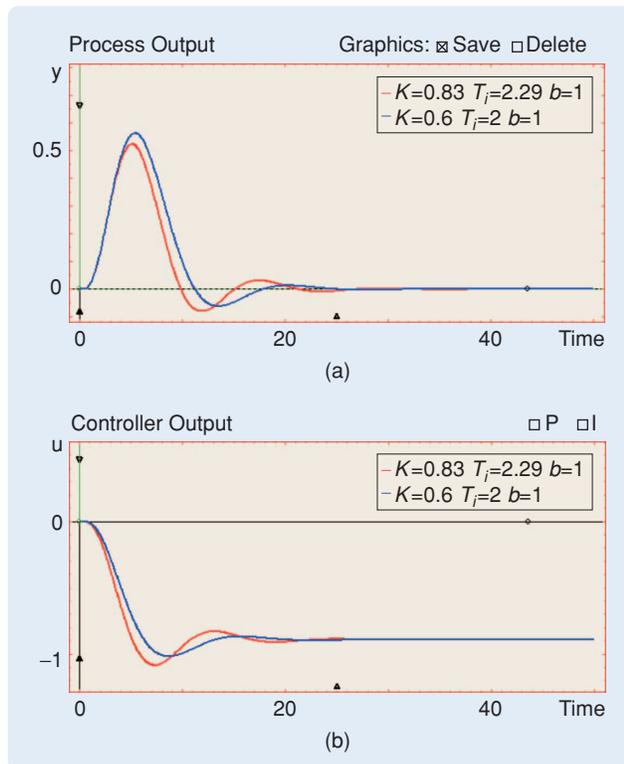
### Settings Menu

The Settings menu of PID Basics is divided into six groups. Arbitrary transfer functions can be selected using the first entry, Process Transfer Function. The numerator and denominator are introduced using a Matlab form.

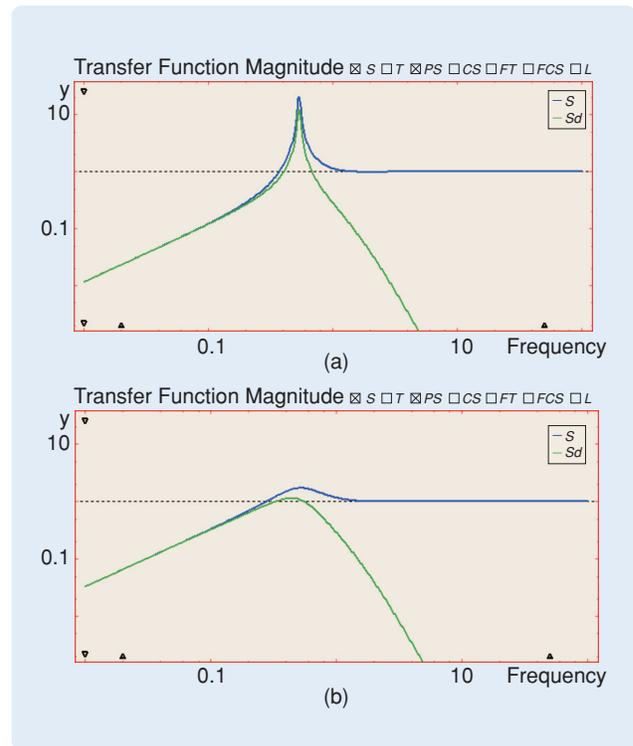
Specific values for controller parameters can be entered using the Controller Parameters menu. Time and frequency responses can be selected from the third entry, Time/Frequency Domain, which has the options Time Domain, Frequency Domain, and Both Domains. The results can be stored and recalled using the Load/Save menu, which has the options Save Design and Load Design. All data on the screen can be saved using the option Save Report. From the menu selection Simulation the user can modify the simulation time, change the maximal time delay to avoid slow simulations, and activate the Sweep option to show the results for several controller parameters simultaneously. Parameters are swept between specified limits. This option is available only in the time-domain mode. When active, new radio buttons appear in the controller-parameters zone to permit the selection of the desired parameter to be swept. The last menu option, Examples Advanced PID Book, loads examples from [8], which the user can explore by modifying parameters.

### Analysis and Control Design for Load Disturbances

Load disturbances are typically low-frequency signals that drive the system away from its desired behavior. The



**FIGURE 5** Load-disturbance response and influence of the integral gain  $k_i$ . For a system with  $P(0) \neq 0$  and a controller with integral action, the low-frequency approximation is  $G_{yd} \approx sP(0)/k_i$ , where  $k_i = K/T_i$  is the integral gain. For load disturbances with low-frequency content, the integral gain  $k_i$  is a measure of load-disturbance attenuation. The (a) process outputs and (b) control signals to load disturbances, respectively, are shown for two PI controllers with  $k_i$  values of 0.36 (in red) and 0.30 (in blue). The controller with larger integral gain provides a faster response to load disturbances.



**FIGURE 6** Frequency-domain interpretation of the load-disturbance response. Figure 5 shows that high values of the integral gain  $k_i$  provide better response to load disturbances. Although this rule is true, it must be used carefully. The frequency-domain responses of  $G_{yd}$  and  $S$  for two PI controllers with (a)  $k_i = 0.85$  and (b)  $k_i = 0.30$ , respectively. As can be seen, large values of  $k_i$  imply large peaks of the sensitivity function  $S = 1/(1 + PC)$ . Therefore, a tradeoff occurs between load disturbance rejection and robustness.

response to load disturbances is a key issue in process control, since most controllers attempt to keep process variables close to desired setpoints [9]. The following example shows the effects of load disturbances and the influence of the controller parameters. The setpoint and noise amplitudes are set to zero, and the load disturbance is set to 0.9 at  $t = 0$ . The process transfer function is given by  $G(s) = (s + 1)^{-4}$ . The response of the process variable to load disturbances is given by the transfer function

$$G_{yd} = \frac{P}{1 + PC} = PS = \frac{T}{C}.$$

If  $P(0) \neq 0$  and the controller has integral action, then the low-frequency approximation is  $G_{yd} \approx sP(0)/k_i$ , where  $k_i = K/T_i$  is the integral gain. For load disturbances with low-frequency content, the integral gain  $k_i$  is a measure of load-disturbance attenuation. Figure 5 shows the load-disturbance responses for two PI controllers with  $k_i$  given by 0.36 (in red) and 0.30 (in blue). Although the controller with larger integral gain provides faster response and smaller values for IAE and  $\epsilon_{max}$  to load disturbances, the stability margins are reduced. Figure 6 shows the frequency responses of  $G_{yd}$  and  $S$  for two PI controllers with large and small values of  $k_i$  (0.85 and 0.30, respectively). This figure reflects that large values of  $k_i$  imply large peaks of the sensitivity function. Therefore, a tradeoff becomes necessary between load-disturbance rejection and robustness.

Some tuning methods allow a tradeoff between robustness and load disturbance response. The approximate M-constrained integral-gain optimization (AMIGO) method [8], [10]–[12] maximizes integral gain under a robustness constraint; see “AMIGO Design Method.” The result of applying AMIGO to this example is shown in Figure 7. The AMIGO-step method is used to design a PI controller with  $K = 0.414$  and  $T_i = 2.66$ . The response to load disturbances is slower than the results presented in Figure 5, but stability margins result are improved, with  $M_s = 1.32$  and  $M_t = 1$ .

## PID LOOP SHAPING

This section briefly describes the main aspects of PID Loop Shaping. The main screen of the tool is shown in Figure 8.

### Process

The process transfer function can be selected and modified depending on the option selected from the Settings menu.

## AMIGO Design Method

Although load disturbances are often the major consideration in process control, robustness and measurement noise must also be considered. Requirements on setpoint response can be dealt with separately by using a controller with two degrees of freedom. The Ziegler-Nichols rules for tuning PID controllers are especially influential. These rules, however, have severe drawbacks, since they use insufficient process information and can yield closed-loop systems with poor robustness [11]. Loop shaping [13] can also be used for PID control, which gives a flexible design method that allows a tradeoff between performance and robustness. The design approach maximizes the integral gain subject to constraints on the maximum sensitivity. This method is called M-constrained integral gain optimization (MIGO) [8], [11].

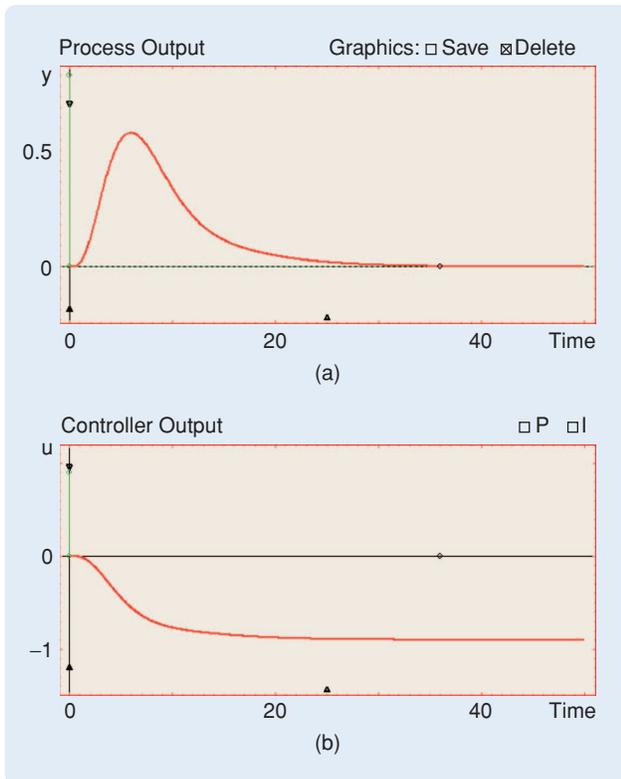
AMIGO (approximate MIGO) design, which is a tuning method in the spirit of Ziegler and Nichols, is the result of finding simple tuning rules for the MIGO method. A large batch of representative processes is selected, including a wide variety of systems with essentially monotone step responses that are typically encountered in process control. Controllers for each process in the batch are then obtained by applying the MIGO design. Having obtained the controller parameters, correlations with normalized process parameters are found by deriving the AMIGO tuning rules. Tables S1 and S2 show these tuning rules for PI and PID controllers in the time and frequency domains. Analysis of these rules can be found in [8]. The main feature of this design method is that it facilitates tradeoffs between robustness and performance. The method thus focuses on load disturbances by maximizing the integral gain and adding a robustness constraint.

**TABLE S1 Time-domain AMIGO tuning rules for first-order time delay (FOTD) models.  $L$  represents time delay,  $T$  is the time constant, and  $K_p$  is the static gain of the process.  $K$ ,  $T_i$ , and  $T_d$  are proportional gain, integral time, and derivative time parameters of PID controllers.**

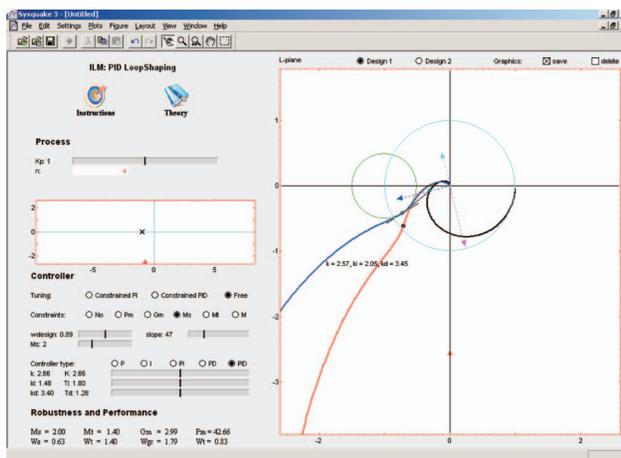
Controller	$K$	$T_i$	$T_d$
PI	$\frac{15}{K_p} + \left(0.35 - \frac{LT}{(L+T)^2}\right) \frac{T}{K_p L}$	$0.35L + \frac{13LT^2}{T^2 + 12LT + 7L^2}$	–
PID	$\frac{1}{K_p} (0.2 + 0.45 \frac{T}{L})$	$\frac{0.4L + 0.8T}{L + 0.1T} L$	$\frac{0.5LT}{0.3L + T}$

**TABLE S2 Frequency-domain tuning rules.  $K_{180}$  is the process gain value at frequency  $\omega_{180}$ ,  $T_{180} = (2\pi)/\omega_{180}$  is the corresponding period, and  $\kappa = K_{180}/K_p$  is the gain ratio.  $K$ ,  $T_i$ , and  $T_d$  are proportional gain, integral time, and derivative time parameters of PID controllers.**

Controller	$K$	$T_i$	$T_d$
PI	$\frac{0.16}{K_{180}}$	$\frac{T_{180}}{1 + 4.5\kappa}$	–
PID	$(0.3 - 0.1\kappa^4)/K_{180}$	$\frac{0.6}{1 + 2\kappa} T_{180}$	$\frac{0.15(1-\kappa)}{1 - 0.95\kappa} T_{180}$



**FIGURE 7** Load disturbance response for a PI controller using the approximate M-constrained integral gain optimization (AMIGO) step method. This method enables the compromise described in Figure 6, focusing on load disturbances by maximizing integral gain and adding a robustness constraint. The (a) process output and (b) control signal to load disturbances, respectively, for a PI controller designed using AMIGO and with  $K = 0.414$  and  $T_i = 2.66$ . The slow response compared with Figure 6 corresponds to increased stability margins.



**FIGURE 8** The user interface of the module PID Loop Shaping, showing both Free and Constrained PID tuning. The loop transfer function is shown for two designs under the Free design option. Proportional, integral, and derivative action are manipulated directly by drawing the arrows. In the Constrained PID, tuning the target point is constrained to lie on the sensitivity circle.

Several process models are available, and their parameters can be modified using sliders as described in PID Basics. In addition, a free transfer function can be selected (menu option Interactive TF), where poles and zeros can be defined graphically as shown in Figure 8.

### Controller

The Controller part of the tool shows the various parameters and properties of PID Loop Shaping to perform loop shaping. The design point of the process transfer function is determined at a specified frequency  $\omega$ . The design point is shown by a green circle on the L-plane graphic. The corresponding point of the loop transfer function at the frequency  $\omega$  is called the target point.

The controller used in PID Loop Shaping is parameterized as

$$C(s) = k + \frac{k_i}{s} + k_d s,$$

which yields the loop transfer function

$$L(s) = C(s)P(s) = kP(s) + \left(\frac{k_i}{s} + k_d s\right)P(s).$$

The point on the Nyquist curve of the loop transfer function corresponding to the frequency  $\omega$  is given by

$$L(i\omega) = kP(i\omega) + i\left(-\frac{k_i}{\omega} + k_d \omega\right)P(i\omega). \quad (4)$$

PID Loop Shaping provides three methods for tuning the parameters to move the process transfer function from the design point to the target point. These methods are listed in the Tuning zone as Free, Constrained PI, and Constrained PID. Free tuning allows an unconstrained loop to be shaped by dragging on the control parameters. Constrained PI and Constrained PID permit the calculation of the controller parameters based on some constraints on the target point. That is, the focus can be placed on how the loop transfer function changes when controller parameters are modified, which reveals the parameter values required to obtain a given shape of the loop transfer function. For PI and PD control the mapping is uniquely given by one point. For PID control it is also possible to obtain an arbitrary slope  $\vartheta$  of the loop transfer function at the target point. When the Free tuning option is selected, sliders are used to modify the controller gains  $k$ ,  $k_i$ , and  $k_d$ , as shown in Figure 8. The controller gains can also be changed by dragging arrows, as illustrated in the same figure. From (4), the proportional gain changes  $L(i\omega)$  in the direction of  $P(i\omega)$ , the integral gain  $k_i$  changes  $L(i\omega)$  in the direction of  $-iP(i\omega)$ , and the derivative gain  $k_d$  changes  $L(i\omega)$  in the direction of  $iP(i\omega)$ .

For the Constrained PI and Constrained PID tuning options, the target point can be limited to move on the unit circle, the sensitivity circles, or the real axis. In this

way loop shaping is enabled with specifications on gain and phase margins or on the sensitivities. In the case of Constrained PI it is necessary to find controller gains providing the desired target point. Dividing (4) by  $P(i\omega)$  and separating the real and imaginary parts gives

$$k = \Re\left(\frac{L(i\omega)}{P(i\omega)}\right), \quad (5)$$

$$-\frac{k_i}{\omega} + k_d\omega = \Im\left(\frac{L(i\omega)}{P(i\omega)}\right) = A(\omega). \quad (6)$$

With  $k_d = 0$ , (5) and (6) yield the two parameters of the PI controller.

An additional condition is required for the Constrained PID tuning option. Hence, it is observed that

$$\begin{aligned} L'(s) &= C'(s)P(s) + C(s)P'(s) \\ &= C'(s)P(s) + \frac{L(s)P'(s)}{P(s)} \\ &= \left(-\frac{k_i}{s^2} + k_d\right)P(s) + \frac{L(s)P'(s)}{P(s)}. \end{aligned} \quad (7)$$

The slope of the Nyquist curve is then given by

$$iL'(i\omega) = i\left(\frac{k_i}{\omega^2} + k_d\right)P(i\omega) + iC(i\omega)P'(i\omega). \quad (8)$$

The complex number represented by (8) has the phase angle  $\vartheta$  if

$$\Im(iL'(i\omega)e^{-i\vartheta}) = 0. \quad (9)$$

Results (7)–(9) imply that

$$\frac{k_i}{\omega^2} + k_d = \frac{\Re\left(L(i\omega)\frac{P'(i\omega)}{P(i\omega)}e^{-i\vartheta}\right)}{\Re(P(i\omega)e^{-i\vartheta})} = B(\omega). \quad (10)$$

Combining (10) with (5)–(6) gives the controller parameters

$$k_i = -\omega A(\omega) + \omega^2 B(\omega), \quad (11)$$

$$k_d = \frac{A(\omega)}{\omega} + B(\omega), \quad (12)$$

where  $A(\omega)$  and  $B(\omega)$  are given by (6) and (10), respectively.

The design frequency  $\omega$  can be chosen using the slider `wdesign` or graphically by dragging the green circle on the process Nyquist curve (black curve in Figure 8). The target point on the Nyquist plot and its slope can be dragged graphically. The slope can also be changed using the slider slope. Furthermore, it is possible to constrain the target point using the Constraints radio buttons to the unit circle

(Pm), the negative real axis (Gm), circles representing constant sensitivity (Ms), constant complementary sensitivity (Mt), or constant sensitivity combinations (M). When sensitivity constraints are active, the associated circles are drawn in the L-plane plot, and sliders can be used to modify their values. The circles are defined in Table 1.

Figure 8 illustrates designs for two PID controllers and a given sensitivity. The target point is moved to the sensitivity circle, and the slope is adjusted so that the Nyquist curve is outside the sensitivity circle. The red design shows a PID controller using Free tuning, while the blue design shows a Constrained PID tuning. Specifications that cannot be reached are indicated in the tool by giving the integral or derivative gain negative values in these cases.

### Robustness and Performance Parameters

Robustness and Performance parameters are displayed on the screen below the controller parameters (Figure 8), and these parameters characterize robustness and performance in the same manner as in PID Basics. The values are maximal sensitivity (Ms), sensitivity-crossover frequency (Ws), maximal complementary sensitivity (Mt), complementary sensitivity-crossover frequency (Wt), gain margin (Gm), gain-crossover frequency (Wgc), phase margin (Pm), and phase-crossover frequency (Wpc).

### L-Plane Graphic

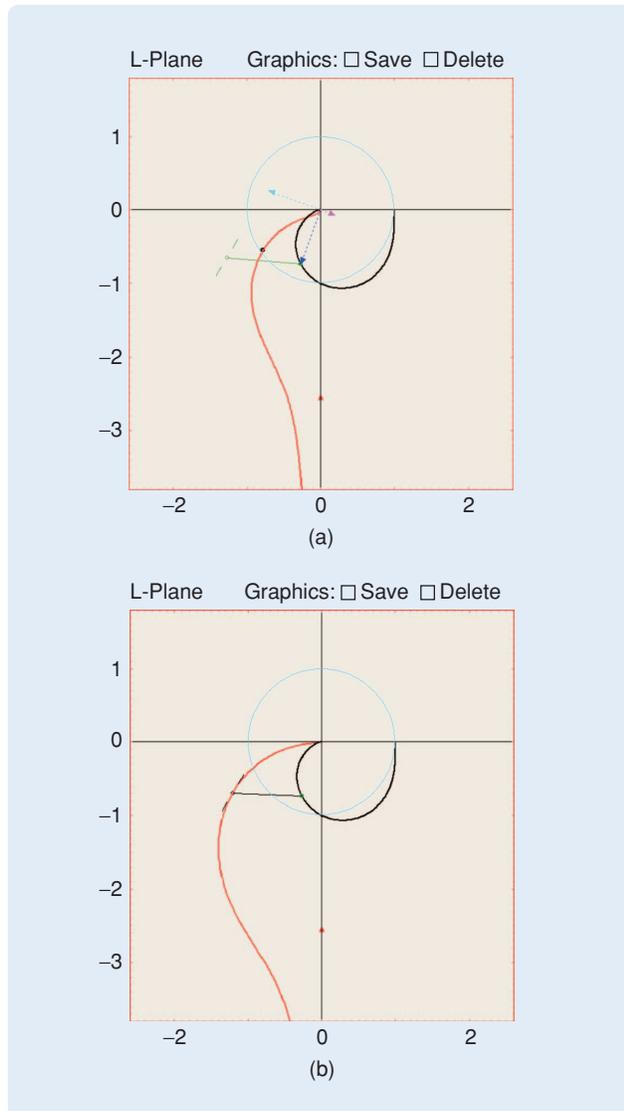
The L-plane graphic is given in the right-hand side of the PID Loop Shaping menu, as shown in Figure 8. This graphic contains the Nyquist plots of the process transfer function  $P(s)$  in black and the loop transfer functions  $L(s) = P(s)C(s)$  in red. Three different views can be shown depending on the tuning options. Figure 9 shows two views, the left one for Free tuning and the right one for Constrained PID tuning. A third view is shown in Figure 8, where two designs are shown simultaneously. The design and target points can be modified interactively on this graphic. The design point is shown in green on the Nyquist curve of the process. The target point is represented in light green in the case of Free tuning and in black for constrained tuning, as shown in Figure 8. The slope of the target point can also be changed

**TABLE 1 Sensitivity circles. This table describes the center and radius of circles that define the loci for constant sensitivity  $M_s$ , constant complementary sensitivity  $M_t$ , constant mixed sensitivity, and equal sensitivities  $M = M_s = M_t$  [8].**

Contour	Center	Radius
$M_s$ -circle	-1	$1/M_s$
$M_t$ -circle	$-\frac{M_t^2}{M_t^2-1}$	$\frac{M_t}{M_t^2-1}$
$M$ -circle	$-\frac{x_1+x_2}{2}$	$\frac{x_1-x_2}{2}$
	$x_1 = \max\left(\frac{M_s+1}{M_s}, \frac{M_t}{M_t-1}\right)$	$x_2 = \max\left(\frac{M_s-1}{M_s}, \frac{M_t}{M_t+1}\right)$

interactively. For **Free** tuning, the controller gains are shown as arrows in the Nyquist plot. The controller gains can be modified interactively by dragging the ends of the arrows. Figures 8 and 9 show examples of these arrows. The scale of the graphic can be changed using the red triangle located at the bottom of the vertical axis.

As noted above, it is possible to impose constraints on the target point. The graphical representation of the target point is modified depending on the constraint selected, restricting



**FIGURE 9** The L-plane graphic. The Nyquist plots of the process transfer function  $P(s)$  (black line) and the loop transfer function  $L(s) = P(s)C(s)$  (red line) are shown. (a) An example of the Free tuning design. The controller gains can be changed by dragging arrows, the proportional gain changes  $L(i\omega)$  in the direction of  $P(i\omega)$  (blue arrow), the integral gain  $k_i$  changes  $L(i\omega)$  in the direction of  $-iP(i\omega)$  (cyan arrow), and the derivative gain  $k_d$  changes  $L(i\omega)$  in the direction of  $iP(i\omega)$  (magenta arrow). (b) An example of the Constrained PID tuning design. In this case, once the user moves the target point (black circle), the controller parameters are calculated using (5)–(12).

its value based on its meaning. Options **save** and **delete** can be found above the L-plane graphic. These options have the same meaning as in PID Basics, making it possible to save designs to perform comparisons. Once the save option is active, two pictures appear, one of which shows the current design in red while the other shows the current design in blue (see Figure 8). Modifications of the controller parameters affect the current (active) design, which can be changed using the options **Design 1** and **Design 2**, which appear on the top of the L-plane graphic. Once a design is chosen, the associated curve is switched to red, and the controller zone is modified based on that design. The controller gain values can be seen by moving the cursor on the curves.

### Settings Menu

The **Settings** menu, which is available in the main menu of PID Loop Shaping, is divided into four groups, following the same structure as in PID Basics. The first entry, called **Process Transfer Function**, is used to choose between several predefined transfer functions or to include a user-specified transfer function through two options. The **String TF** option allows a transfer function to be entered symbolically. For instance,  $P(s) = 1/\cosh \sqrt{s}$  can be represented as  $P = '1/\cosh(\text{sqrt}(s))'$ . Results can be stored and recalled using the **Load/Save** menu. The option **Save Report** can be used to save all essential data in text format, which is useful for documenting results. Specific values for control parameters can be entered with **Parameters** menu option. As in PID Basics, the last menu option (**Examples Advanced PID Book**) allows loading examples from [8].

### Examples

Some of the capabilities of PID Loop Shaping are illustrated by the following examples.

#### Effect of Controller Parameters

The purpose of this example is to illustrate how the Nyquist plot of the loop transfer function changes when the controller parameters are modified.

Consider the process  $P(s) = 1/(s + 1)^4$ . When a P controller is used, the proportional gain changes the loop transfer function  $L(i\omega) = kP(i\omega)$  in the direction of  $P(i\omega)$ . Figure 10(a) shows the effect of modifying  $L(i\omega)$  using a P controller with gain  $k = 2$  (blue curve) and  $k = 2.6$  (red curve). These curves show how the proportional gain modifies the Nyquist plot of the process (black curve) at the frequency  $\omega$  (green circle on the black curve) in the direction of  $P(i\omega)$ . Figure 10(b) shows the same study for an I-controller with  $k_i = 1$  (red curve) and  $k_i = 0.6$  (blue curve). It can be seen that the integral gain  $k_i$  changes  $L(i\omega)$  in the direction  $-iP(i\omega)$ . The derivative gain has the same effect in the direction of  $iP(i\omega)$ . When a PI or PD controller is used, the compensated point at the frequency  $\omega$  is calculated as the sum of two vectors, namely, the proportional vector and the integral or derivative vector. Examples of this capability are

shown in Figure 10(c) and (d), where the process is controlled by a PI controller ( $k = 2.3$  and  $k_i = 0.7$ ) and a PD controller ( $k = 2.1$  and  $k_d = 3.35$ ), respectively.

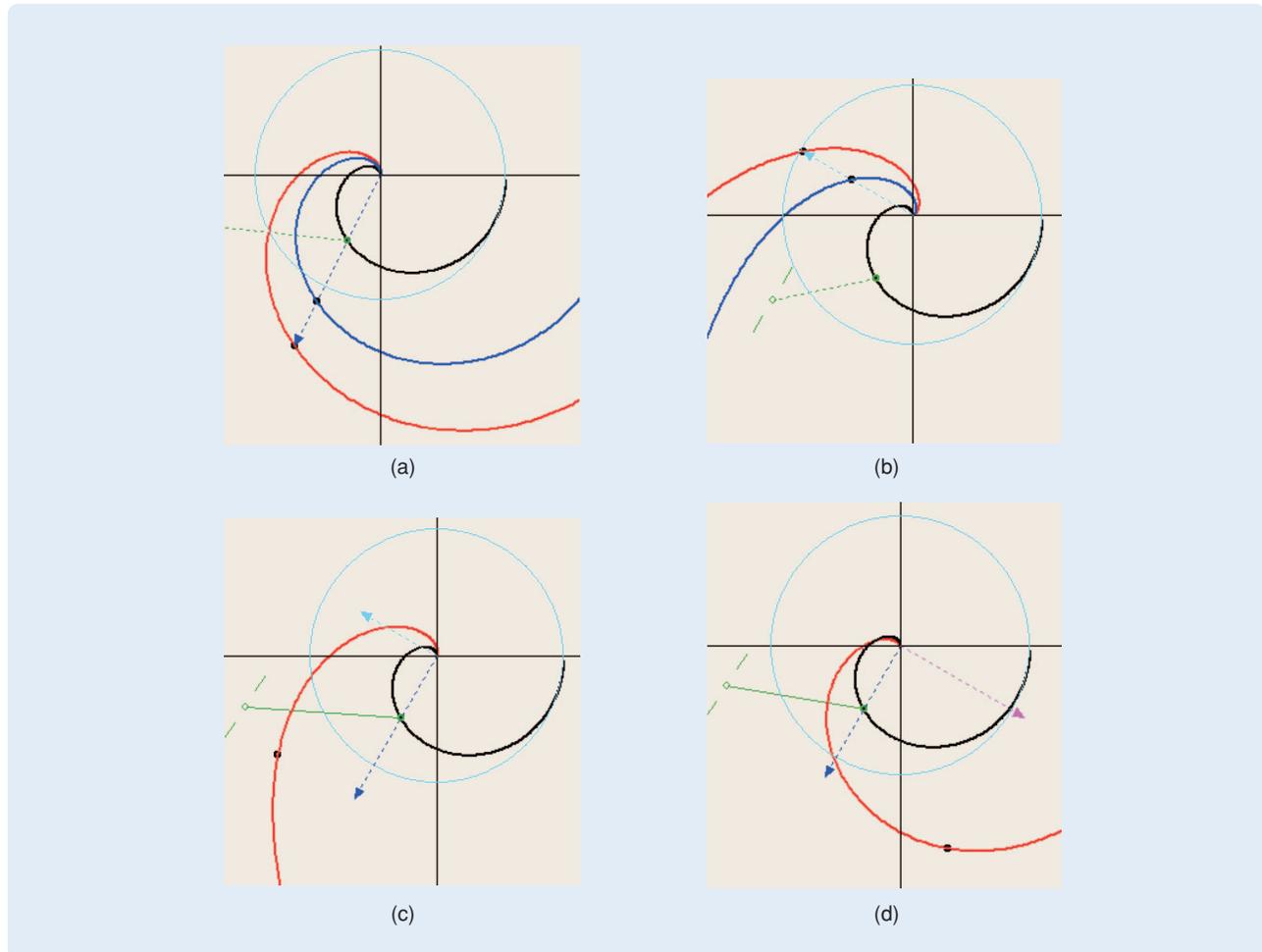
Simple exercises can be used to provide training in loop shaping. For instance, with the above process, it is instructive to calculate the gain for a proportional controller for which the closed-loop system changes from stable to unstable. Before using PID Loop Shaping, the result can be calculated analytically, which yields

$$\begin{aligned} \angle L(i\omega) &= \angle C(i\omega)P(i\omega) = -180, \\ k \frac{1}{(i\omega + 1)^4} &= -180, \quad \omega = 1. \\ |L(i\omega)| &= |C(i\omega)P(i\omega)| = |-1 + 0i|, \\ \left| k \frac{1}{(i\omega + 1)^4} \right| &= -1, \quad k = 4. \end{aligned}$$

PID Loop Shaping can be used to verify the result interactively, as shown in Figure 11. This exercise challenges

students and encourages them to make observations while relating theory to images to develop a broader and deeper understanding.

On the other hand, free interactive designs can also be performed to compare the results with other design methods. For instance, PID Loop Shaping can be used to design a PID controller interactively for the process  $P(s) = 1/(s + 1)^4$ , where the maximal sensitivity value  $M_s$  must be less than 1.5. A PID controller that satisfies this constraint is obtained when  $k = 0.92$ ,  $T_i = 1.8$ ,  $k_i = 0.5$ ,  $T_d = 1.03$ , and  $k_d = 0.95$ . The AMIGO-frequency method can also be used for design, and the results can be compared. The resulting controller is given by  $k = 1.2$ ,  $T_i = 2.48$ ,  $k_i = 0.48$ ,  $T_d = 0.93$ , and  $k_d = 1.12$ . Figure 12 shows the Nyquist plots and time responses using PID Basics for both designs, in blue for the free PID controller and in red for the AMIGO method. The resulting values of  $M_s$  are 1.49 for free PID and 1.46 for the AMIGO method.

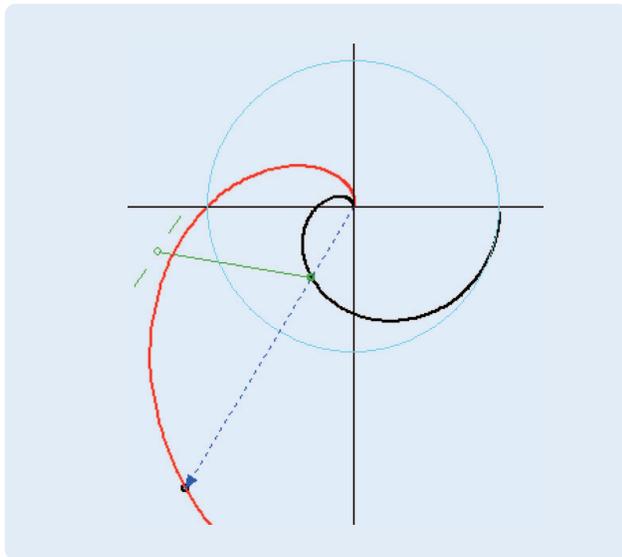


**FIGURE 10** Nyquist plot modifications depending on the controller type. (a) P controller, (b) I controller, (c) PI controller, and (d) PD controller. (a) The modification of  $L(i\omega)$  in the direction of  $P(i\omega)$  using a P controller with gain  $k = 2$  (blue curve) and  $k = 2.6$  (red curve). The same study for an I-controller is shown in (b) with  $k_i = 1$  (red curve) and  $k_i = 0.6$  (blue curve), where  $L(i\omega)$  is modified in the direction  $-iP(i\omega)$ . (c), (d) PI or PD controllers are used, respectively. In these cases, the compensated point at the frequency  $\omega$  is calculated as the sum of two vectors, namely the proportional vector and the integral or derivative vector.

### Effect of the Target Point

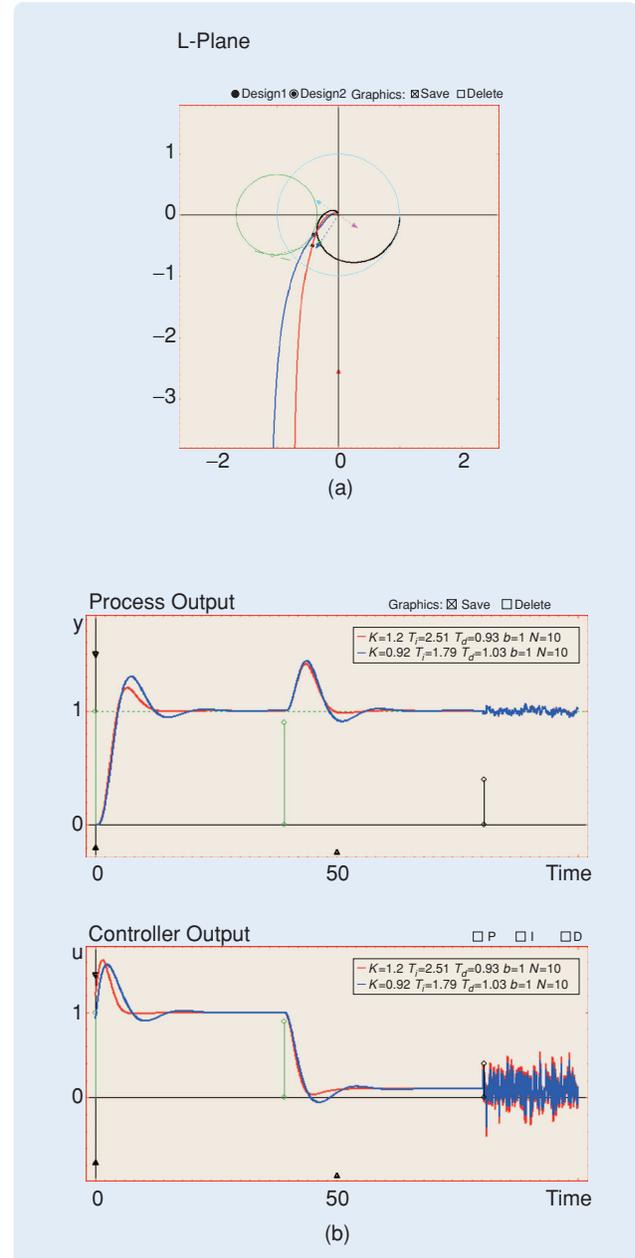
The target point on the Nyquist plot can be reached using an unconstrained design by selecting the **Free** option. The controller gains are interactively adjusted as shown in the free tuning example. Another approach is to use (5)–(12), where the controller gains are calculated after the target point is defined. As discussed above, the target point can be fixed or constrained in various ways, either at any point, to specific values for phase margin and gain margin, or to maximal values of the sensitivity functions. Figure 13 shows an example in which the target point is set to the point  $-0.5 - 0.5i$ . Two constrained designs are shown for the design frequency  $\omega = 0.6$  rad/s. The red curve represents a system compensated by a constrained PID with  $k = 1.32$ ,  $k_i = 1.02$ , and  $k_d = 2.15$ , while the blue curve represents a constrained PI with  $k = 1.32$  and  $k_i = 0.15$ . Although both controllers reach the target point, better results are obtained for the PID controller because the slope can be freely adjusted (the value for this example is  $\vartheta = 22$ ). The PID controller provides better robustness properties with  $M_s = 1.45$ ,  $k_i = 1.02$ ,  $G_m = 5.32$ , and  $P_m = 40.15$ , versus a PI controller with  $M_s = 1.83$ ,  $k_i = 0.15$ ,  $G_m = 2.69$ , and  $P_m = 75.77$ .

Similar examples can be used to restrict the target point for phase margin, gain margin, or maximal values of the sensitivity functions. Figure 14(a) shows an example where a combined sensitivity constraint is required for  $M_s \leq 2$  and  $M_t \leq 2$ . This constraint is fulfilled in two different ways, namely, by using Constrained PID (red curve) and Constrained PI (blue curve). Another example combining sensitivity function and gain margin constraints is shown in Figure 14(b), with the specification that the gain margin be equal to three and  $M_s \leq 2$ . These specifications are



**FIGURE 11** Stability limit on the critical point  $-1 + 0i$ . A typical example for presenting loop shaping is to search for the lowest gain that makes the system unstable. This task can be interactively performed with PID Loop Shaping as shown in this figure.

established by maximizing the integral gain  $k_i$ . Hence, the constraint gain margin is chosen, and the target point is located in such a way that  $G_m = 3$ . Then, a Constrained PID controller is selected, where the design point and the slope are modified until  $M_s \leq 2$  and the integral gain is maximized. The final controller is given by  $k = 1.38$ ,  $k_i = 0.52$ , and  $k_d = 0.54$  for  $\omega = 1.02$  and  $\vartheta = 32$ .



**FIGURE 12** Example of loop shaping with  $M_s < 1.5$ . PID Loop Shaping can be used to compare various designs. In this figure, (a) Nyquist plots and (b) time-domain responses generated with PID Basics are shown to compare an unconstrained design ( $k = 0.92$ ,  $T_i = 1.8$ ,  $k_i = 0.5$ ,  $T_d = 1.03$ , and  $k_d = 0.95$ ) with an alternative design developed using the AMIGO-frequency method ( $k = 1.2$ ,  $T_i = 2.48$ ,  $k_i = 0.48$ ,  $T_d = 0.93$ , and  $k_d = 1.12$ ).

### The Derivative Cliff

We again consider the process transfer function  $P(s) = 1/(s + 1)^4$ . It is desirable to maximize the integral gain  $k_i$  subject to the robustness constraint  $M_s \leq 1.4$ . The resulting controller has the parameters  $k = 0.925$ ,  $k_i = 0.9$ , and  $k_d = 2.86$ , where the Nyquist plot of the loop transfer function is shown in red in Figure 15(a). It can be seen that the Nyquist curve has a loop, called a derivative cliff. As explained in [8], this feature, which is due to excessive controller phase lead, results from having a PID controller with complex poles, which occurs when  $T_i < 4T_d$ . In this example the relation is  $T_i = 0.33T_d$ . Figure 15(b) shows, in red, the time response of the controller, which yields oscillatory outputs. For comparison, the results for a controller with  $T_i = 4T_d$  are shown in blue in Figure 15(a) and (b) with the controller parameters  $k = 1.1$ ,  $k_i = 0.36$ , and  $k_d = 0.9$ . The responses for this controller are improved, despite larger overshoot in response to load disturbances. This example is available in the Settings menu of PID Loop Shaping.

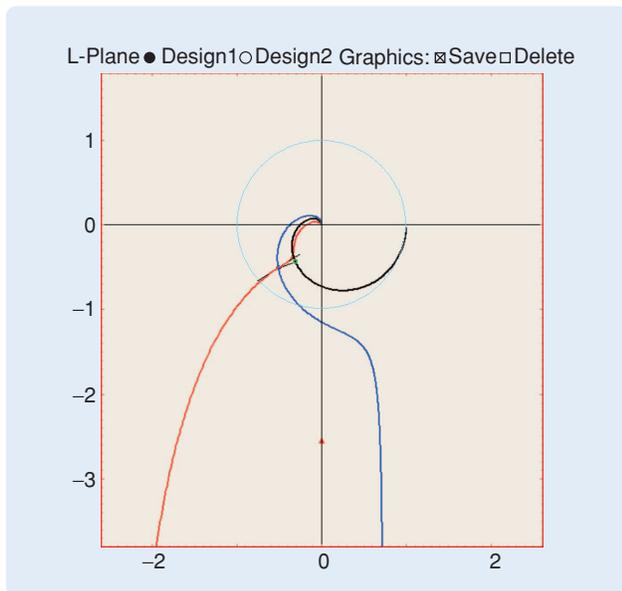
### PID WINDUP

The purpose of the PID Windup module is to facilitate understanding of integral windup and a method for compensating it [8]. For a control system with a wide range of operating conditions, it may happen that the control variable reaches the actuator limits. When this situation occurs in loops using a controller with integral action, the feedback loop is broken and the integral may reach large val-

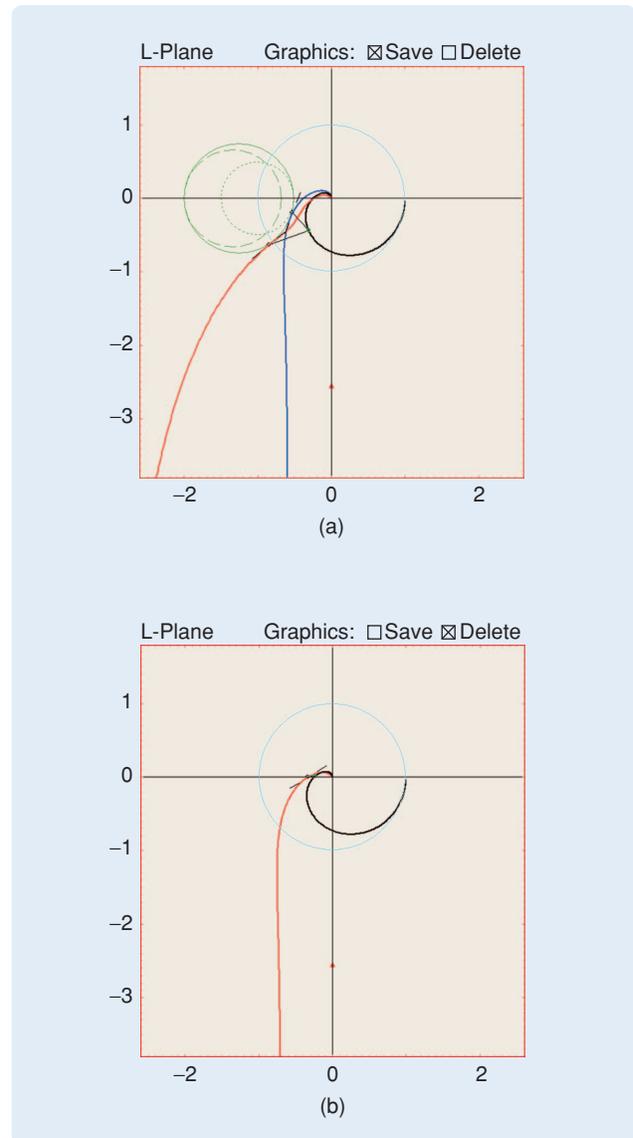
ues, maintaining the control signal saturated for a long time, resulting in large overshoot, and undesirable transients. This problem is known as windup phenomenon [8].

Windup can be avoided in different ways. Back calculation and tracking [8] is illustrated in the block diagram in Figure 16. The system remains unchanged when the saturation is not active. However, when saturation occurs, the integral term in the controller is modified until the control signal is out of the saturation limit. This modification is not performed instantaneously but dynamically with a time constant  $T_i$  called the tracking time constant [8].

The module PID Windup shows process outputs and control signals for unlimited control signals, limited

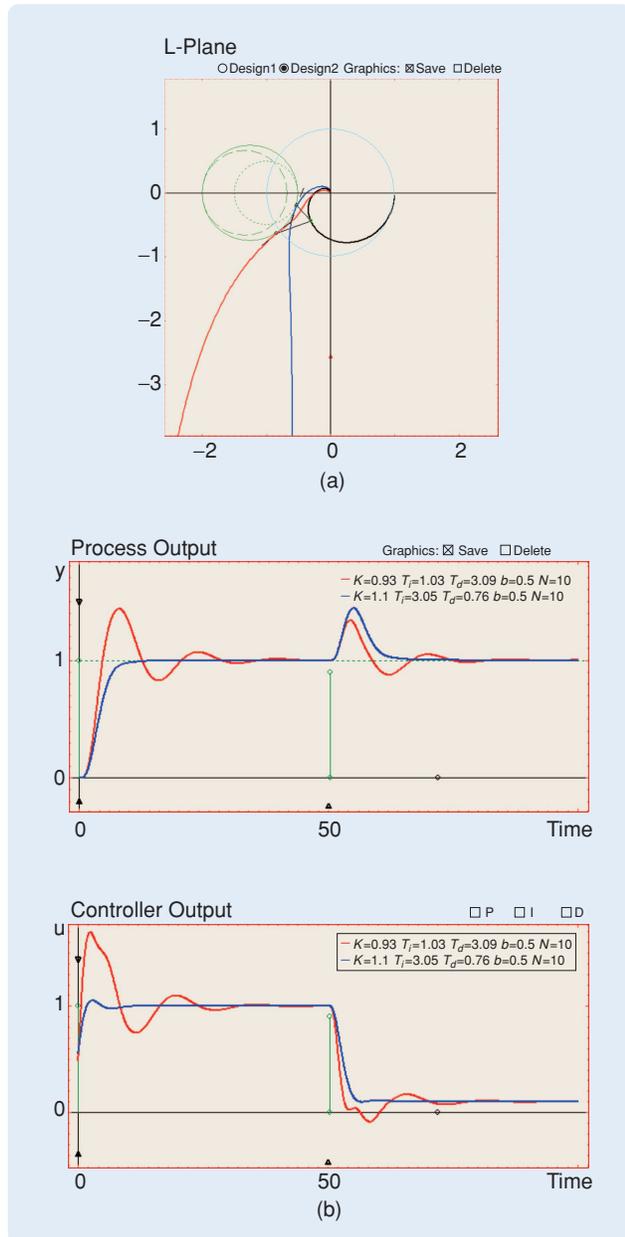


**FIGURE 13** Example of a constrained design with a target point of  $-0.5 - 0.5i$ . The target point can be constrained to reach arbitrary specifications. Once a point is constrained, the controller parameters are automatically calculated. This plot shows PI ( $k = 1.32$ ,  $k_i = 0.15$ ) and PID ( $k = 1.32$ ,  $k_i = 1.02$ ,  $k_d = 2.15$ ) controllers, both reaching the target point. The PID controller provides better results due to the use of the slope as an allowable third degree of freedom as described in (11) and (12), where the slope  $\vartheta$  takes the value 22.



**FIGURE 14** Example of a constrained design with sensitivity and gain-margin constraints. These plots show an example where the target point is constrained to reach specified values for the (a) combined sensitivity functions with  $M_s \leq 2$  and  $M_t \leq 2$ , and (b) gain margin with limited sensitivity values with  $Gm = 3$  and  $M_s \leq 2$ .

control signals without antiwindup, and limited control signals with antiwindup. The user interface is shown in Figure 17. Process models and controller parameters can be selected in the same way as in the other modules. The saturation limits of the control signal can be determined either by entering the values or by dragging the lines in the saturation scheme.



**FIGURE 15** Derivative cliff example. (a) Nyquist plot and (b) time-domain responses. This example shows that optimization of  $k_f$ , which is aimed at fulfilling robustness specifications, can provide controllers with excessive phase lead, as represented by the loop in the red curve. This behavior is a consequence of the presence of complex zeros due to  $T_i < 4T_d$ . The same example is shown in blue for  $T_i = 4T_d$ , where this problem is avoided [8].

## The Interactive Tool

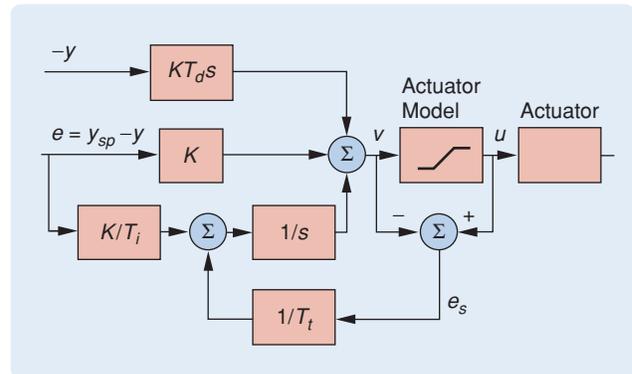
We now describe the main aspects of PID Windup.

### Process

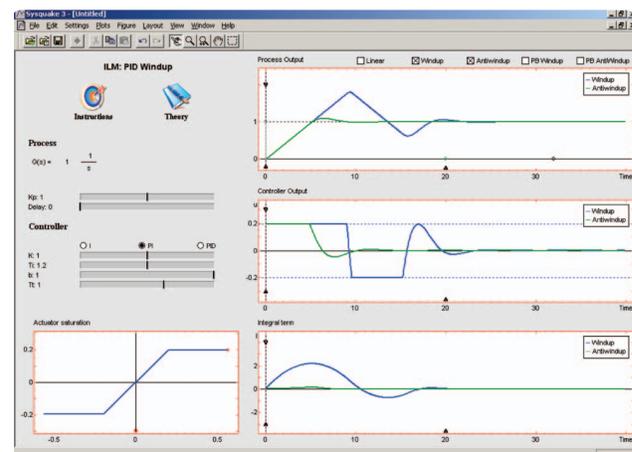
The Process area is similar to that described in PID Basics and PID Loop Shaping. The time delay is modified using a slider instead of a text edit, so that the time delay effect on the antiwindup mechanism can be analyzed.

### Controller

The Controller area contains information about the controller parameters and actuator saturation. Three kinds of



**FIGURE 16** PID controller with antiwindup scheme, where  $K$  is the controller proportional gain,  $T_i$  is the controller integral time,  $T_d$  is the controller derivative time,  $y_{sp}$  is the setpoint,  $y$  is the process output,  $e$  is the tracking error,  $v$  is the controller output,  $u$  is the saturated controller output, and  $e_s$  is the difference between the controller output  $v$  and the saturated controller output  $u$ . In this scheme the control signal remains unconstrained when the saturation is not active. When saturation occurs, the integral control action is modified until the control signal is out of the saturation limit. The modification of the integral element is performed dynamically by adjusting the tracking time constant  $T_i$  [8].



**FIGURE 17** The user interface of the module PID Windup, showing the windup phenomenon and application of the antiwindup technique. Several graphical elements are used to interactively analyze typical problems and solutions associated with windup. The example shown in the figure illustrates the windup phenomenon (in blue) and the result of applying the antiwindup technique (in green).

controllers with integral action can be selected (I, PI, PID), where several sliders are available to change the controller parameters, including the tracking time constant  $T_t$ . A saturation graphic is also available in this zone. The Actuator Saturation graphic allows the saturation limits to be determined by dragging the small red circle located on the upper saturation value. In this graphic, a symmetric saturation is selected for pedagogical purposes.

### Graphics

Time responses for process output, control signal, and integral action are available in three graphics, namely, Process Output, Controller Output, and Integral Term. In the same way as in PID Basics, multiple interactive graphical elements can be used to change the setpoint, load disturbance, measurement noise, or horizontal and vertical scales (see Figure 17). These graphics can simultaneously represent the controlled system in linear, nonlinear with windup, and nonlinear with antiwindup modes. These representations can be configured using the checkboxes located above the Process Output graphic. For instance, Figure 17 shows an example containing the nonlinear with windup and nonlinear with antiwindup modes.

The dotted pink vertical line in Figure 18 is helpful for comparing the outputs of the different plots at the same time instant. The saturation limits can be altered using the

dotted blue horizontal lines available in the Controller Output graphic (see Figure 17).

The notion of proportional band is useful for understanding the windup effect, and is included in PID Windup. The proportional band is defined as the range of process outputs such that the controller output is in the linear range  $[y_{\min}, y_{\max}]$ . For a PI controller, the proportional band is limited by

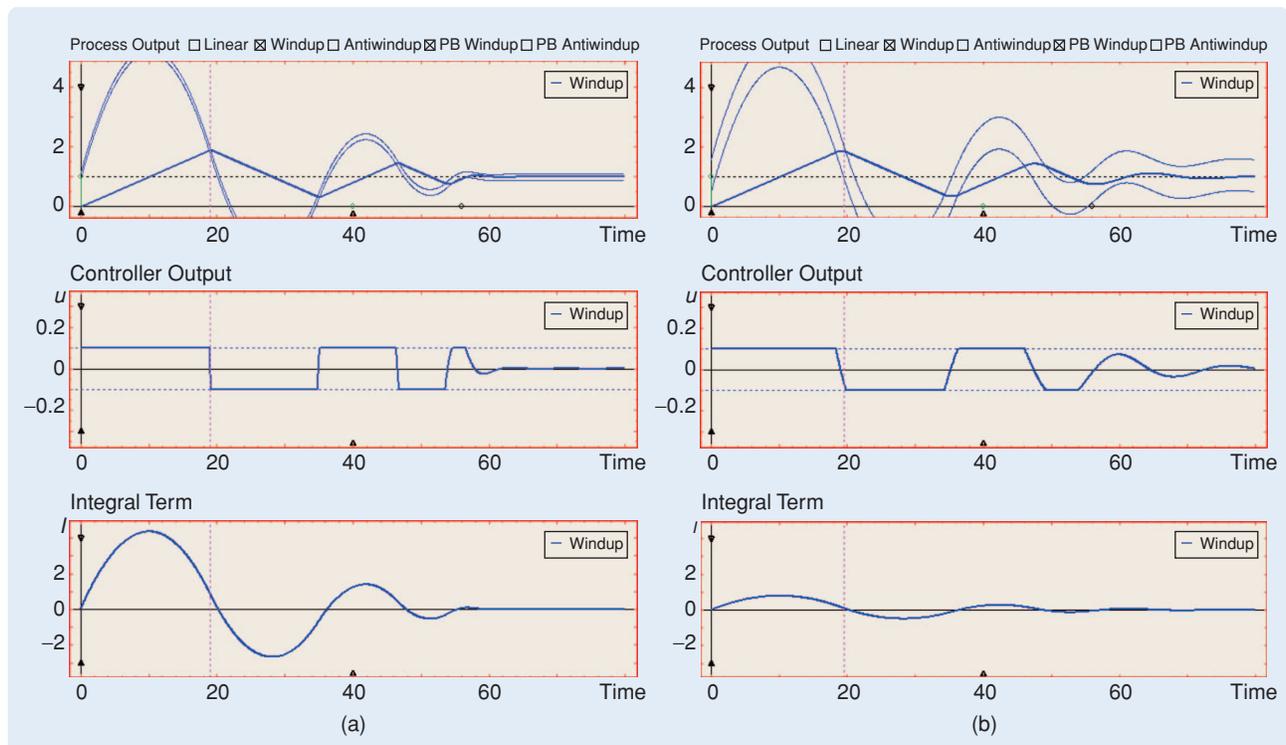
$$y_{\min} = by_{sp} + \frac{I - u_{\max}}{K}, \quad (13)$$

$$y_{\max} = by_{sp} + \frac{I - u_{\min}}{K}, \quad (14)$$

where  $I$  is the integral term of the controller, and  $u_{\max}$  and  $u_{\min}$  are the control signal limits.

Expressions (13) and (14) hold for PID control when the proportional band is defined as the band where the predicted output  $y_p = y + T_d(dy/dt)$  is in the proportional band  $[y_{\min}, y_{\max}]$ . The proportional band has the width  $(u_{\max} - u_{\min})/K$ , and is centered around  $by_{sp} + I/K - (u_{\max} + u_{\min})/(2K)$ .

Two additional checkboxes called PB Windup and PB Antiwindup, appear near the top of plot Process Output. The activation of these options shows the proportional bands for the windup and antiwindup cases in the Process Output graphic. The proportional bands are shown as dotted green and blue curves, respectively, as shown in Figure 18.



**FIGURE 18** Example of the windup phenomenon with proportional band for (a)  $K = 1$  and (b)  $K = 0.4$ . In [8] the notion of proportional band is described as being a useful tool for understanding the effects of windup. The proportional band is an interval such that the actuator does not saturate when the instantaneous value of the process output or its predicted value is inside this band. These plots show two examples demonstrating how the control signal is saturated when the process output is inside the band shown in blue. The interactive pink line of the graphics can be used to test this idea.

## Settings Menu

The Settings menu has the same structure as in PID Basics and PID Loop Shaping. The process transfer function can be chosen from the entry Process Transfer Functions, and numerical values of the parameters can be introduced using Controller Parameters. Essential data and results can be saved and recalled using the Load/Save menu options. The menu selection Simulation makes it possible to choose the simulation time and activate the Sweep option, which can be used to show the results for several values of the tracking time constant. Several examples from [8] can be loaded from the Examples entry.

## Examples

The following examples illustrate properties of the PID Windup module.

### Understanding the Windup Phenomenon

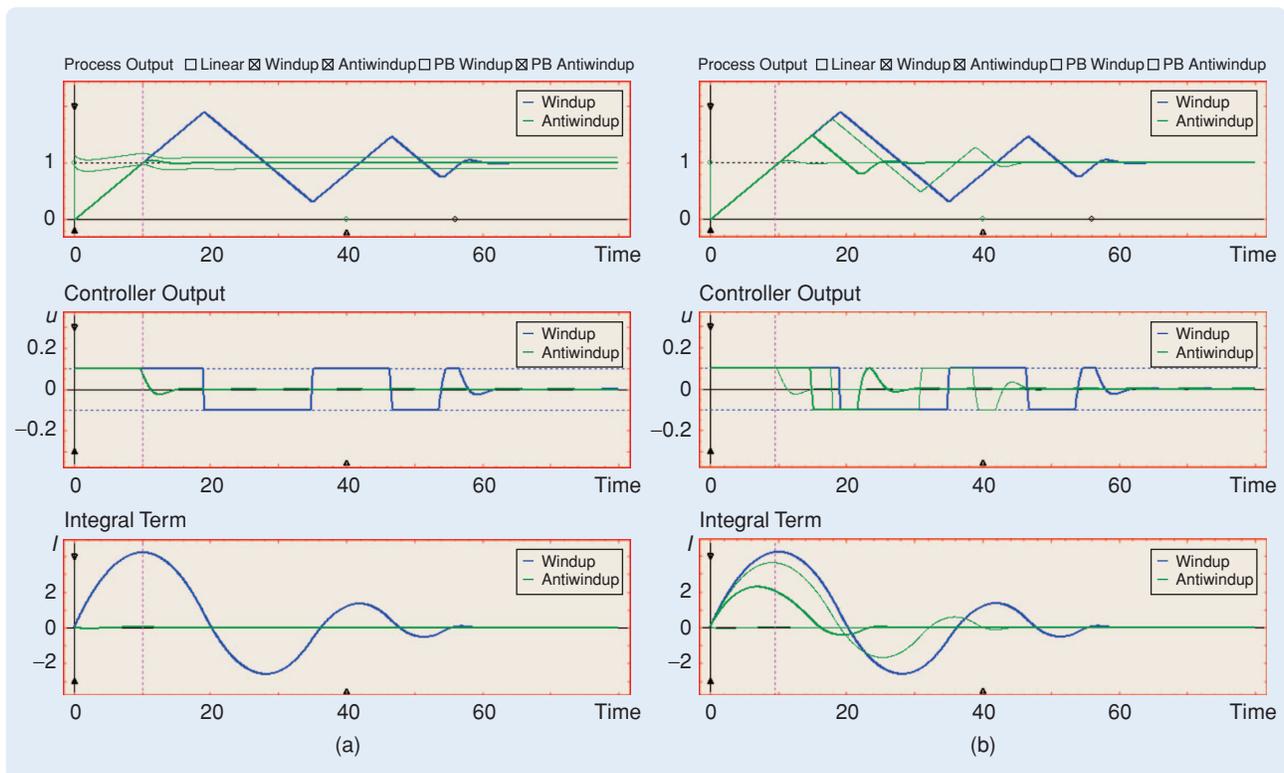
Windup can be studied using the first entry from the Examples option menu. This example from [8] uses the pure integrator process  $P(s) = 1/s$  controlled by a PI controller with parameters  $K = 1$ ,  $T_i = 1.2$ , and  $b = 1$ , and with the control signal limited to  $\pm 0.2$ . Figure 17 shows the time responses for this example. The control signal is saturated from  $t = 0$ . The process output and the integral term increase while the control error is positive. Once the

process output exceeds the setpoint, the control error becomes negative, however the control signal remains saturated due to the large value of the integral term. The time responses are shown in Figure 17.

The proportional band can be drawn in this example using the PB Windup checkbox shown in Figure 18(a). Using the vertical line, the user can see that the process output remains inside the band while the control signal is working in linear mode and outside the proportional band when the control signal is saturated. Large controller gains provide narrow proportional bands, with more energetic control signals and therefore longer saturation times, while small controller gains give wider proportional bands. Figure 18(b) illustrates this effect, where the proportional controller gain is reduced to 0.4, producing a wider proportional band.

### Antiwindup

The process  $P(s) = 1/s$  is also useful for visualizing the antiwindup technique. The same controller parameters, namely,  $K = 1$ ,  $T_i = 1.2$ ,  $b = 1$ , are used, and the tracking time constant is set to  $T_t = 1$ . Figure 19(a) shows the responses for both cases control with and without antiwindup. The system with antiwindup remains in saturation for only a short period of time, with the magnitude of the integral term considerably reduced. The proportional



**FIGURE 19** Example of the effect of the tracking time constant  $T_t$  on in the antiwindup technique. (a) Antiwindup and (b) effect of  $T_t$ . These plots show the results of applying the antiwindup technique to the example shown in Figure 17. The integral signal is considerably reduced, allowing the control signal to remain in saturation during a shorter period of time. The proportional band for the antiwindup technique is shown in green. The process output remains inside the band most of the time.

band for the PI controller with antiwindup is shown in the same figure. It can be seen that the proportional band is wider than for PI without antiwindup [Figure 19(a)], where the process output remains most of the time. The effect of the tracking time constant is illustrated in Figure 19(b) for  $T_t = 0.1, 10, 50$ . In this scenario, the Sweep menu option is used. High values of  $T_t$  make the antiwindup too slow to be effective, while low values reset the integral term quickly with improved results. It may thus seem advantageous to always have small values of  $T_t$ . However, the next example shows some situations where this choice is not advisable.

### The Tracking Time Constant

The tracking time constant is an essential parameter because it determines the reset rate for the integral term of the controller. It may seem advantageous to have a small value for this constant. However, measurement errors may accidentally reset the integral term when the tracking time constant is too small. The following example illustrates this phenomenon, when a measurement error occurs in the form of a short pulse. The transfer function of the process is

$$P(s) = \frac{1}{(0.5s + 1)^2},$$

and the controller is a PID controller with  $K = 3.5$ ,  $T_i = 0.52$ ,  $T_d = 0.14$ ,  $N_d = 10$ ,  $b = 1$ , and  $T_t = 1$ .

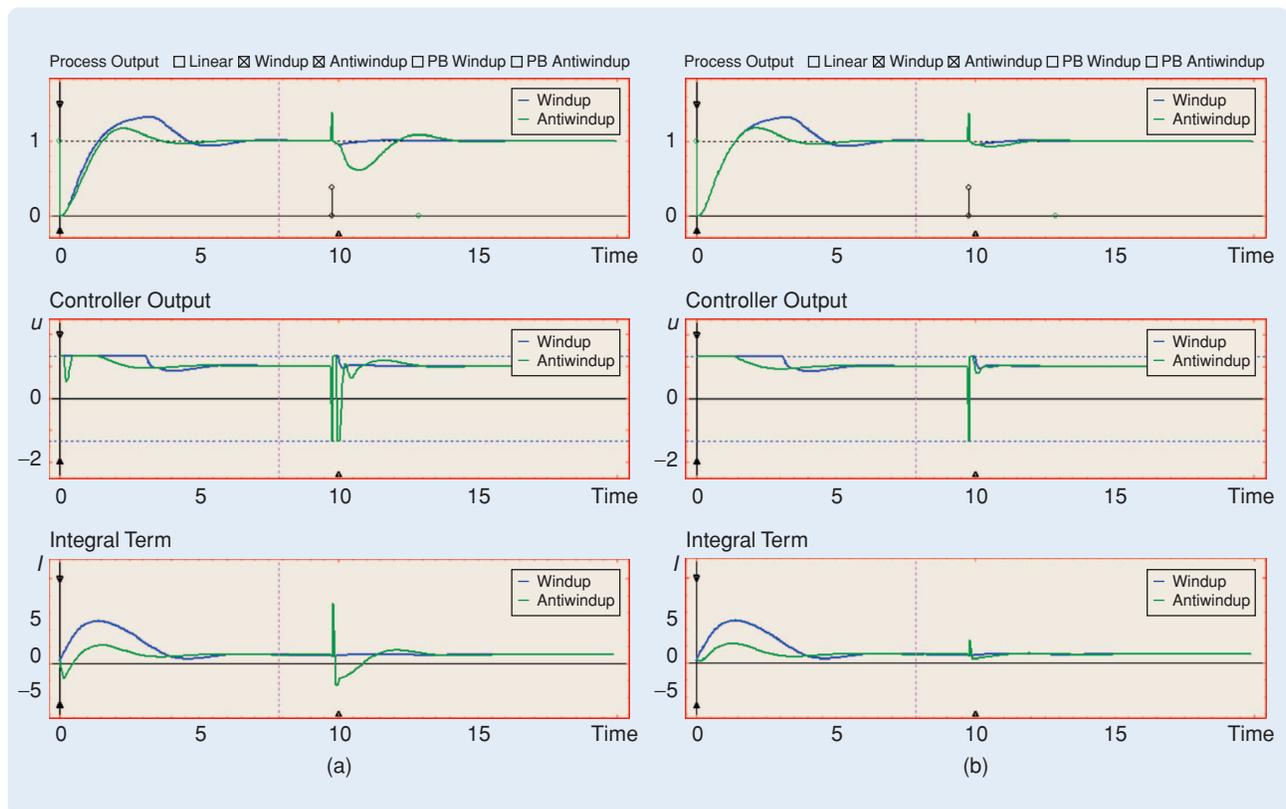
Figure 20(a) shows the control results. A large transient appears after the pulse, and the integral term is excessively reduced.

Various rules are suggested in [8] for choosing the tracking time constant. One choice is  $T_t = (T_i + T_d)/2$ . Figure 20(b) shows an example with  $T_t = (T_i + T_d)/2 = 0.33$ , where the response is considerably improved.

### CONCLUSIONS

In this work a set of interactive modules that comprise ILM-PID is presented to support the teaching and learning of basic automatic control concepts. These tools are intended mainly to include interactivity in the visual content of [8]. The modules focus on PID control, studying feedback fundamentals from the standpoint of the time and frequency domains, including robustness issues, measurement of noise filtering, load-disturbance rejection, and windup phenomenon.

The importance of interactivity in automatic control education has been shown in the context of teaching and learning. In the authors' experience, interactivity offers excellent support to education and learning by enhancing the motivation and participation of future



**FIGURE 20** Tuning the tracking time. (a) Reset by measurement noise and (b) tuning using rules. (a) illustrates the disadvantage of using a short tracking time constant. The short pulse disturbance at time  $t = 10$  results in excessive reduction of the integral term and a large disturbance in the process output. In (b) the choice is  $T_t = (T_i + T_d)/2$ .

engineers. The interactive learning modules developed in this work are freely available from the authors [7] to test these interactive features in control education and professional training.

## REFERENCES

- [1] M. Johansson, M. Gäfvert, and K.J. Åström, "Interactive tools for education in automatic control," *IEEE Contr. Syst. Mag.*, vol. 18, no. 3, pp. 33–40, 1998.
- [2] J.L. Guzmán, M. Berenguel, and S. Dormido, "Interactive teaching of constrained generalized predictive control," *IEEE Contr. Syst. Mag.*, vol. 25, no. 2, pp. 79–85, 2005.
- [3] J. Sánchez, S. Dormido, and F. Esquembre, "The learning of control concepts using interactive tools," *Comput. Appl. Eng. Educ.*, vol. 13, no. 1, pp. 84–98, 2005.
- [4] J.L. Guzmán, "Interactive control system design," Ph.D. dissertation, University of Almería, Spain [Online]. Available: <http://www.ual.es/~joguzman/PhdThesisGuzman.pdf>
- [5] Y. Pignet, *SysQuake 3 User Manual*. Lausanne, Switzerland: Calerga Sàrl, 2004 [Online]. Available: <http://www.calerga.com>
- [6] J.L. Guzmán, K.J. Åström, S. Dormido, T. Häggglund, and Y. Pignet, "Interactive learning modules for PID control," in *Proc. 7th IFAC Symp. Advances in Control Education*, 2006, Madrid, Spain [Online]. Available: <http://aer.ual.es/ilm/>
- [7] ILM Web site, *Interactive Learning Modules*, 2005 [Online]. Available: <http://www.calerga.com/contrib/1/index.html>
- [8] K.J. Åström and T. Häggglund, *Advanced PID Control*. Research Triangle Park, NC: Instrum. Soc. Amer. 2005.
- [9] F.G. Shinskey, *Process-Control Systems. Application, Design, and Tuning*. New York: McGraw Hill, 1996.
- [10] T. Häggglund and K.J. Åström, "Revisiting the Ziegler-Nichols tuning rules for PI control," *Asian J. Contr.*, vol. 4, no. 4, pp. 364–380, 2002.
- [11] T. Häggglund and K.J. Åström, "Revisiting the Ziegler-Nichols step response method for PID control," *J. Process Contr.*, vol. 14, no. 6, pp. 635–650, 2004.
- [12] T. Häggglund and K.J. Åström, "Revisiting the Ziegler-Nichols tuning rules for PI control—Part II, the frequency response method," *Asian J. Contr.*, vol. 6, no. 4, pp. 469–482, 2004.
- [13] K.J. Åström, H. Panagopoulos, and T. Häggglund, "Design of PI controllers based on non-convex optimization," *Automatica*, vol. 34, no. 4, pp. 585–601, 1998.

## AUTHOR INFORMATION

**José Luis Guzmán** (joguzman@ual.es) earned a computer science engineering degree in 2002 and a Ph.D. in 2006, both from the University of Almería, Spain, where he is an assistant professor and a researcher in the Automatic Control, Electronics, and Robotics Group. Currently his interests center on control education and robust model predictive control techniques, with applications to interactive tools, virtual and remote labs, and agricultural processes. He can be contacted at the Universidad de Almería, Dpto. de Lenguajes y Computación, Ctra. Sacramento s/n, 04120, Almería, Spain.

**Karl Johan Åström** received his M.Sc. in engineering physics in 1957 and his Ph.D. in control and mathematics in 1960 from the Royal Institute of Technology in Stockholm, Sweden. After graduating, he worked for IBM Research for five years. In 1965, he became professor at Lund Institute of Technology/Lund University, where he founded the Department of Automatic Control. He is now emeritus at Lund University and a part-time visiting professor at the University of California at Santa Barbara. He is a Life Fellow of the IEEE. He has broad interests in control and its applications. He has received many awards, including the IEEE Medal of Honor and the Quazza medal from IFAC.

**Sebastián Dormido** holds a degree in physics from the Complutense University in Madrid, Spain (1968) and a Ph.D. from the University of the Basque Country, Spain (1971). In 1981 he was appointed professor of control engineering at the Universidad Nacional de Educación a Distancia. His activities include computer control of industrial processes, model-based predictive control, robust control, and model and simulation of continuous processes. He has authored and coauthored more than 150 technical papers in international journals and conferences. From 2002 to 2006 he was president of the Spanish Association of Automatic Control CEA-IFAC.

**Tore Häggglund** received his M.Sc. in engineering physics in 1978 and his Ph.D. in automatic control in 1984, both from Lund Institute of Technology, Sweden, where he is currently a professor. Between 1985 and 1989 he worked for Alfa Laval Automation (now ABB) on the development of industrial adaptive controllers. He is a coauthor of *Advanced PID Control* and *Process Control in Practice*. His current research interests are in the areas of process control, tuning and adaptation, and supervision and detection.

**Manuel Berenguel** is a professor at the University of Almería, Spain. He earned an industrial engineering degree and Ph.D. from the University of Seville, Spain, where he has been a researcher and associate professor for six years. His research interests are in predictive, adaptive, and robust control, with applications to solar energy systems, agriculture, and biotechnology. He has authored and coauthored more than 100 technical papers in international journals and conferences and is coauthor of the book *Advanced Control of Solar Plants* (Springer, 1997).

**Yves Pignet** is CEO of Calerga Sàrl, a company he co-founded in 2001 to develop and commercialize scientific software such as Sysquake, of which he is the main developer. He received a diploma in microtechnics in 1991 and a Ph.D. degree in control systems in 1997, both from EPFL, Lausanne, Switzerland. His research interests include robust control, real-time controllers and their use in mobile robotics, and interactive software and its applications.

