Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

A Software Tool for Robust PID Design

Olof Garpinger, Tore Hägglund

Department of Automatic Control,
Lund University, Box 118, SE-221 00 Lund, Sweden
(Tel: +46 46 222 03 62; e-mail: olof.garpinger@control.lth.se)

Abstract: This paper presents a fast, interactive and easily modifiable software tool for robust PID design. The Matlab based program is supposed to give people with moderate knowledge on PID control a possibility to learn more and also be a future part of an autotuner. The PID design is made by minimizing the integrated absolute error value during a load disturbance on the process input. The optimization is performed with H_∞ constraints on the sensitivity and complementary sensitivity function, providing a robust closed loop system. Nelder Mead optimization is used with the AMIGO method providing an initial controller. The proposed method works well, and is very efficient, on a large batch of systems common in process industry. The design tool is also shown to work on a highly oscillatory process model.

Keywords: Algorithms and software, Autotuning, Disturbance rejection, PID Control, Process Control, Robust control

1. INTRODUCTION

The PID controller is by far the most common controller in industry today. Even so, a lot of these controllers are poorly tuned. Two of the main reasons for that is lack of knowledge and time among the operators. As a consequence, many controllers are set to default values. In other cases, the derivative part is turned off because it was not used correctly, giving noisy signals. It would therefore be a good idea to educate the operators in the possibilities of the PID controller and to provide them with simple and fast design tools. This paper describes a program that achieves both goals and should be useful for people in the industry as well as for academics.

There are many PID design methods available today and some of the most famous are collected and analysed in Åström and Hägglund (2005). Of these, the MIGO and AMIGO methods (also see Panagopoulos et al. (2002) and Hägglund and Åström (2004)) are probably those most worth mentioning in connection to this paper. They are based on optimization of load disturbance rejection under robustness constraints. A further development of the MIGO method, and largely based on the same method as used in this paper, was presented in Nordfeldt (2005). An advantage with Nordfeldt's method is that it also works for some more advanced process structures. This paper focuses on the software that solves the optimization problem and how it can be used to increase people's understanding of PID control.

The proposed PID control design method is incorporated in several Matlab functions. There are many good reasons to have a software based tool for control design and analysis. In Åström and Hägglund (2001) it is pointed out that it would be of great value to have software that can give persons with moderate knowledge on PID controllers a possibility to experiment on those and at the same time be able to use the program to build controllers for a real plant, by incorporating it into an autotuning procedure. For simulation experiments and real use purposes, the presented software is able to provide a well working controller with analysis tools in just a few seconds time. The advanced user should also be able to modify the optimization problem to broaden the possibilities. Besides the proposed program, which is intended to be free of charge and downloadable, there are already several commercial software packages able to provide PID design tools using a variety of methods. Many of these are collected in Li et al. (2006) and another one with very similar features to the proposed is presented in Oviedo et al. (2006).

2. DESIGN CRITERION

The proposed PID controller design tool is mainly meant to work well for systems common in process industry. The kind of plants encountered there are often stable, monotone and primarily affected by low frequency load disturbances.

In order for the controller design to work well on a process, $P(s)$, it is important to take all system signals into consideration, especially if optimization is used. Figure 1 shows a block diagram of the system that the PID controller, $C(s)$, is designed for. There are two external signals entering the system, namely load disturbance d (mainly low frequency) and measurement noise n (assumed high frequency). Of the closed loop transfer functions, those of greatest interest for this paper are the complementary sensitivity function $T(s)$ and the sensitivity function $S(s)$, defined as

$$T(s) = \frac{P(s)C(s)}{1 + P(s)C(s)}, \quad S(s) = \frac{1}{1 + P(s)C(s)}.$$

The PID controller is on parallel form with a second order low pass filter

$$C(s) = K(1 + \frac{1}{sT_i} + sT_d) \cdot \frac{1}{1 + sT_f + (sT_f)^2/2},$$

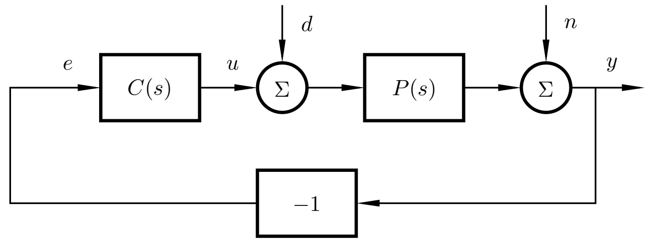

Fig. 1. A load disturbance, d, and measurement noise, n, act on the closed loop system with process $P(s)$ and PID controller $C(s)$.

on the measurement signal. T_f, is chosen to weight the degree of measurement noise rejection.

The objective of the proposed PID design method is to find the PID controller giving the least integrated absolute error (IAE) value when a load disturbance d, modelled as a step, is acting on the closed loop system. The optimization is done under the constraints that the closed loop system is stable and that the open loop Nyquist curve is tangent to one or two prespecified circles in the complex plane without entering either of them (see Figure 2), thus maximizing the gain. These two circles are called the M_s- and M_p-circles, which sizes and positions are given by

$$M_s = \max_\omega |S(i\omega)|, \quad M_p = \max_\omega |T(i\omega)|,$$

hence the names. The resulting, non-convex, optimization problem can be written as

$$\min_{K,T_i,T_d \in \mathcal{R}^+} \int_0^\infty |e(t)|dt = IAE_{load} \tag{1}$$

$$\begin{aligned} \text{subject to } & |G_o(i\omega) - C_{M_s}|^2 \geq R_{M_s}^2 \quad \forall \omega \in \mathcal{R}^+, \\ & |G_o(i\omega) - C_{M_p}|^2 \geq R_{M_p}^2 \quad \forall \omega \in \mathcal{R}^+, \\ & |G_o(i\omega^s) - C_{M_s}|^2 = R_{M_s}^2, \\ & |G_o(i\omega^p) - C_{M_p}|^2 = R_{M_p}^2, \end{aligned}$$

where $e(t)$ is the control error, $G_o(i\omega)$ is the open loop frequency response, ω^s are frequencies for which $G_o(i\omega)$ is tangent to the M_s-circle and vice versa for ω^p on the M_p-circle. Either ω^s or ω^p could be an empty vector, but not at the same time. The radius and centre point of the M_s-circle are denoted by R_{M_s} and C_{M_s} respectively, with corresponding measures for the M_p-circle, R_{M_p} and C_{M_p}. Small M_s- and M_p-values result in large circles. In the software, the maximum allowed M_s- and M_p-values can be prespecified by the user ($M_s = M_p = 1.4$ is default, resulting in 41.8° phase margin). The M_s- and M_p- criterions are known to set the closed loop robustness towards process variations, disturbances and nonlinearities as described in Åström and Hägglund (2005). MIGO on the other hand uses a simplified robustness criterion called the M-circle, defined as the smallest circle that can be drawn around both the M_s- and M_p-circle.

3. ALGORITHM OVERVIEW

The main goal of the new design algorithm was to develop a fast, interactive and easily modifiable software tool for robust PID design.

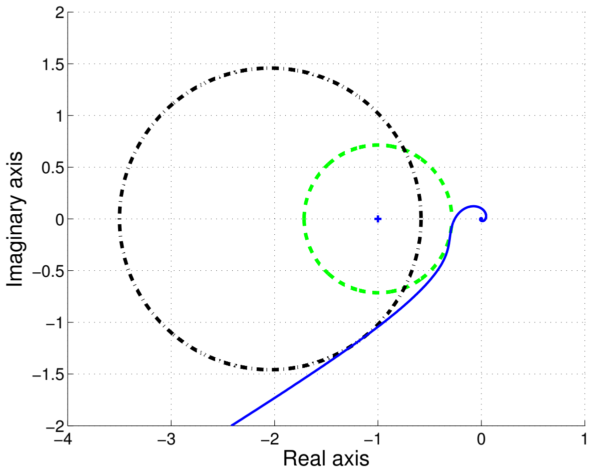

Fig. 2. The M_s-circle (dashed), M_p-circle (dash-dotted) and the open loop Nyquist curve (solid) when the optimization criterions are fulfilled.

A non-convex optimization problem like (1) may have many local minima. It is therefore hard to guarantee that the solution obtained always is the global solution. It is also difficult to draw any general analytical conclusions as the problem is far from trivial. The method of gridding does however give a possibility of drawing surface plots of the cost function. These can be used to determine whether or not it is likely that a given solution is in fact the global minimum. This is also the major reason why gridding is an optional optimization method in the proposed design program.

Analysis of many cost function surfaces have shown that if not all, then at least a lot of them only have one minimum. This finding gave the idea to use a faster and more advanced optimization tool than gridding, called the Nelder Mead (NM) method, Nelder and Mead (1965), in order to find the minimum in the T_i-T_d plane, see below.

The new algorithm can be summarized by

(1) Given a linear transfer function, initial PID parameters are chosen using the AMIGO method.
(2) NM optimization finds the PID controller giving the minimum cost function in the T_i-T_d plane.
    (a) For each T_i-T_d couple, a proportional gain, K, is found such that the constraints are fulfilled.
    (b) Simulink simulations are used to calculate IAE-values in the points through which the NM method proceeds.

An interactive program menu has been added to make it possible for the user to change a number of settings in the algorithm as well as for the presentation of the results. When the program is run in Matlab, the menu will come up unless the opposite is stated by the user. New default values for the optimization can also be set as input parameter. This is especially useful for batch runs, when you may want to choose the settings before a number of program runs are started. An experienced user should easily be able to modify the program, to for instance, change the optimization method or at least to change the cost function.

4. ALGORITHM DETAILS

In this section, the optimization algorithm will be explained in further detail.

4.1 The Nelder Mead Method

Nelder Mead optimization belongs to the subclass of optimization methods called direct search methods. The main theme among these is that they only use function values without creating approximations of the function gradients explicitly. These methods are especially useful if, for instance, the cost to evaluate the function is high and if it is impossible to derive the exact gradient. These statements apply to the optimization problem (1). Whenever the cost function is evaluated, the feasible proportional gains must be calculated and Simulink simulations run. The simulations are particularly costly if the given PID parameters, at a certain grid point, gives a very sluggish closed loop.

The Nelder Mead method is a simplex-based method. There are many papers and books which describe in detail how the NM algorithm works (see for instance Walters et al. (1991) and Lagarias et al. (1998)). Two of the reasons why the method is popular are that it is easy to both understand and implement. It is only necessary to look at two dimensional NM optimization in this paper as (1) can be viewed as an unconstrained minimization problem in $\mathcal{R}^2$, when K is chosen separately. Two dimensional NM optimization can be interpreted as triangle search progression with variable area.

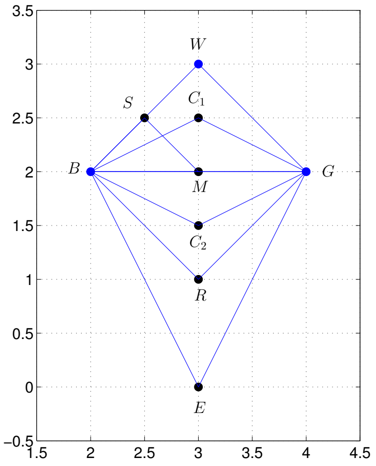

Fig. 3. The Nelder Mead progression in one iteration. The initial simplex is the one with corners in B, G and W. The simplex will change it's shape depending on function evaluations in closely situated points.

In order to begin the NM optimization, an initial triangle has to be specified. The function to be minimized - lets call it f - is evaluated at all three edges and the points are sorted in the order: B (Best, lowest function value $f(B)$), W (Worst, highest function value $f(W)$) and G (Good, function value, $f(G)$, in between the other two). From this point, the algorithm will alter the shape of the triangle to give a new one with less total cost. These steps are well explained in the given sources and will not be presented in further detail here. Figure 3 gives a hint of possible new simplexes. When a new simplex has been determined - again with corners B, G and W - the algorithm will iterate until a termination condition has been fulfilled.

4.2 Initial Values

It is preferable to have a good initial guess of where the minimum is located to have fast convergence of the optimization. Another reason is that there is a chance - although not so big when solving (1) - of ending up in a local minimum. The method used to receive an initial controller in the proposed algorithm is called AMIGO, which is a tool for robust PID (and PI) synthesis. To understand AMIGO, it is also important to understand the MIGO method for PID design.

The optimization problem that the MIGO design deals with is very similar to (1). But instead of minimizing over the IAE-value, it uses the Integrated Error,

$$IE_{load} = \int_0^\infty e(t)dt,$$

as cost function and the M-circle as robustness constraint, to determine the PID parameters. The IE cost is proportional to $1/k_i = T_i/K$, which reduces the problem to maximizing the k_i-gain over the robustness area.

The AMIGO design is basically a set of formulas yielding K, T_i and T_d. In order to determine these, the MIGO method was run on a large number of systems common in process industry (with $M_s = M_p = 1.4$). Secondly, each and every process in the batch was approximated as a first order system with time delay (FOTD)

$$G_p(s) = \frac{K_p}{sT+1}e^{-sL}. \quad (2)$$

The PID-parameters were then plotted versus the normalized time delay, $\tau = L/(L+T)$, and parameter fittings were made on these curves resulting in the formulas.

In the proposed PID design method, the system of interest is approximated as a FOTD system, (2), through a step response test and the AMIGO parameters are then determined. Let the index A denote the AMIGO PID parameters. The AMIGO parameters provided are used as one of the corners, (T_d^A, T_i^A), in the initial Nelder Mead simplex. Taking into account that the evaluation time is usually greater far out in the T_i-T_d plane, the two remaining corners have been set to $(0.4T_d^A, T_i^A)$ and $(T_d^A, 0.4T_i^A)$.

4.3 Determining the proportional gain K

The key idea to find K in every point (T_d, T_i), is to determine all K-values putting the open loop Nyquist curve on a circle in the complex plane (at every frequency point ω), resulting in a function $K(\omega)$. Since the method is numerical, the frequency span is divided into a finite number of points ω_k, $k = 1, 2, \ldots$. In order to determine $K(\omega)$, let us first assume that the open loop frequency response, $G_o(i\omega)$, can be written as

$$G_o(i\omega) = K(X(\omega) + iY(\omega)), \quad (3)$$

where $X(\omega)$ and $Y(\omega)$ are the real and imaginary parts of $G'_o(i\omega) = G_o(i\omega)/K$. From the optimization problem (1) we have the constraint

$$|G_o(i\omega) - C|^2 = R^2, \quad (4)$$

for any circle with center in C and radius R. Using (3) and (4), but changing K to $K(\omega)$, will lead to

$$(K(\omega)X(\omega) - C)^2 + (K(\omega)Y(\omega))^2 = R^2 \Rightarrow \tag{5}$$
$$K(\omega)^2 - \frac{2CX(\omega)}{X(\omega)^2 + Y(\omega)^2}K(\omega) + \frac{C^2 - R^2}{X(\omega)^2 + Y(\omega)^2} = 0.$$

The two solutions correspond to the gains for which the open loop Nyquist curve will cross the front and back side of the circle respectively (see Figure 4)

$$K_{1,2}(\omega) = \frac{CX(\omega) \pm \sqrt{R^2(X(\omega)^2 + Y(\omega)^2) - C^2Y(\omega)^2)}}{X(\omega)^2 + Y(\omega)^2}. \tag{6}$$

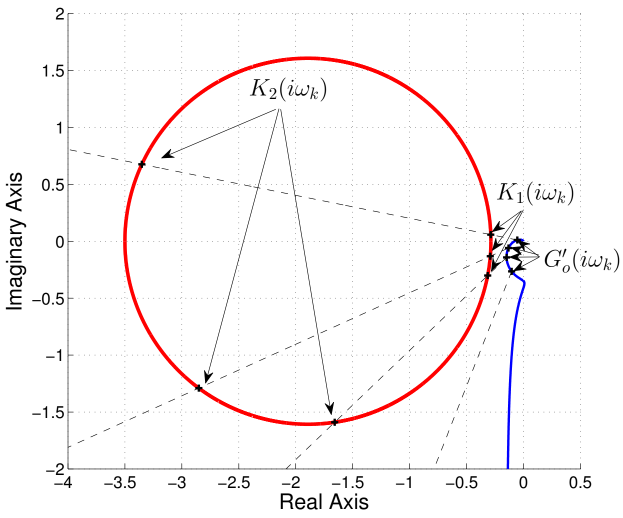

Fig. 4. Proportional gain functions $K_1(\omega_k)$, $K_2(\omega_k)$, for which $K_jG_o'(i\omega_k)$, $j \in [1, 2]$, is tangent to a circle in the complex plane. $G_o'(i\omega_k)$ is the open loop frequency response with $K = 1$ and ω_k denotes the discrete frequency points.

$K_{1,2}(\omega)$ could for instance look like the plots in Figure 5. For some frequency points, (6) will assume imaginary or negative numbers, which are discarded. In the intervals for which K assumes positive real values, there can be multiple minima and maxima.

There are a few observations needed in order to conclude which K-values will fulfill the constraints in (1).

Theorem 1. The open loop Nyquist curve, (3), of an arbitrary controlled process, will be tangent a circle in the complex plane, given by the center point C and radius R, if and only if

$$\frac{dK_1(\omega)}{d\omega} = 0 \quad \text{or} \quad \frac{dK_2(\omega)}{d\omega} = 0 \tag{7}$$

Proof. In vector form, the open loop frequency response becomes

$$G_o(i\omega) = K\begin{pmatrix} X(\omega) \\ Y(\omega) \end{pmatrix}. \tag{8}$$

There are two conditions that has to be fulfilled in order for the open loop Nyquist curve to be tangent to the circle at a given frequency point ω^*. The open loop Nyquist curve should lie on the circle determined by

$$(KX(\omega^*) - C)^2 + (KY(\omega^*))^2 = R^2, \tag{9}$$

while the tangent of the open loop Nyquist curve and the vector between the center point and Nyquist curve should be orthogonal

$$\left(\frac{dG_o(i\omega^*)}{d\omega}\right)^T \begin{pmatrix} KX(\omega^*) - C \\ KY(\omega^*) \end{pmatrix} = 0. \tag{10}$$

Denoting $X'(\omega) = dX(\omega)/d\omega$, $Y'(\omega) = dY(\omega)/d\omega$, (10) can be rewritten as

$$\begin{pmatrix} KX'(\omega^*) \\ KY'(\omega^*) \end{pmatrix}^T \begin{pmatrix} KX(\omega^*) - C \\ KY(\omega^*) \end{pmatrix} =$$
$$K^2X(\omega^*)X'(\omega^*) - KCX'(\omega^*) + K^2Y(\omega^*)Y'(\omega^*) = 0,$$

and in turn, by solving for K, we end up with

$$K = \frac{CX'(\omega^*)}{X(\omega^*)X'(\omega^*) + Y(\omega^*)Y'(\omega^*)}. \tag{11}$$

Let us now go back to equation (5). Taking the derivative with respect to ω, given that $K'(\omega) = dK(\omega)/d\omega$, leaves us with

$$2KK'X^2 + 2K^2XX' - 2CK'X - 2CKX' + \tag{12}$$
$$2KK'Y^2 + 2K^2YY' = 0,$$

with ω omitted. Using $K'(\omega) = 0$, results in

$$K(\omega)X(\omega)X'(\omega) - CX'(\omega) + K(\omega)Y(\omega)Y'(\omega) = 0 \Rightarrow$$
$$K(\omega) = \frac{CX'(\omega)}{X(\omega)X'(\omega) + Y(\omega)Y'(\omega)}, \tag{13}$$

which is identical to (11) in ω^*. Since (9) is fulfilled for all frequencies in $K(\omega)$, the proof is concluded. □

$G_o(i\omega)$ can, however, be tangent to the circle on both the inside or outside, but still have points within (thus giving an infeasible solution). To explain why, it is a good idea to once again view Figure 4. At a given frequency point ω_k, it is obvious that all proportional gains between $K_1(\omega_k)$ and $K_2(\omega_k)$ will place $G_o(i\omega)$ inside the circle, thus resulting in infeasible K. Looking at Figure 5, this means that only

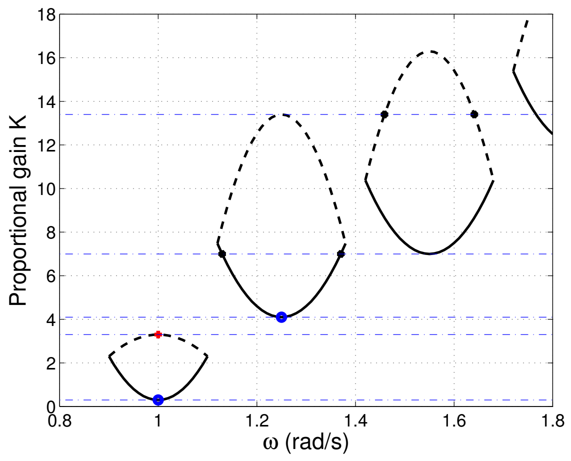

Fig. 5. A constructed sketch of how the functions $K_1(\omega)$ (solid) and $K_2(\omega)$ (dashed) could look for a time delayed system. Only K-values unique in ω - i.e. the first two minima and first maximum in this case - will fulfill the circle constraints in (1).

the minima and maxima, for which K is unique in ω, are feasible. For this particular case, it corresponds to the first two minima and first maximum.

Once all possible K-values have been determined (the two minima and the maximum from Figure 5 for example), closed loop stability is evaluated. If there is stability, the optimal proportional gain at a given point in the T_i-T_d plane, is then given by the K resulting in the lowest IAE-value (determined by Simulink simulations).

Up to now it has been assumed that it is just one circle in the complex plane that the open loop Nyquist curve may be tangent to. Since the constraints of (1) demands that the Nyquist curve is located outside both the M_s- and M_p-circles, the algorithm has to be run twice.

5. EXAMPLES

In this section there will be a few examples highlighting the benefits of the proposed program and algorithm compared to other methods. It will also show that the new method is reliable in many design cases.

Example 1. (The AMIGO test batch). The AMIGO formulas, Hägglund and Åström (2004), were derived using MIGO on a test batch, which includes 134 essentially monotone systems common in process industry. In order to compare with the MIGO PID designs on the batch, the PID design software presented in this paper was modified to use the M-circle as constraint. It took the program just more than one hour to run through all sub-batches except some integrating processes. This gives an average time of 30 seconds per process in the batch. The batch was however run to get a high accuracy on the optimal solution rather than optimized for a fast design. If speed is of essence, the average design time per process could be cut by at least two thirds of the time. The designs was run on an Intel® Dual-Core™, 2.13 GHz with 1 GB RAM and Fedora 7 using Matlab® 7 R2007a. The only two parameters that had to be modified from the default values (depending on the process) in order for the batch to run through properly, were T_f and the frequency grid.

The PID parameters derived by the proposed algorithm were compared to those given by the MIGO method. Since the MIGO method was not derived to minimize IAE, the proposed method should give lower values at all times. The MIGO method is however a good indicator to see if the new algorithm works or not. The batch run showed that the two methods are very similar. In average, the new controllers resulted in IAE values at 95% of what the MIGO controllers gave over the whole batch. This gives both a strong indicator that the new program works properly and that the MIGO method gives essentially IAE minimized controllers for the batch.

To see the benefit of using the M_s- and M_p-circles instead of the M-circle, the whole batch was compared when the two different constraints were used respectively. Figure 6 shows that the biggest percentual gain is given for low values on the normalized time delay, τ, while more delay dominated systems are less dependent on the choice of the constraints. This indicates that the IAE can be decreased a great deal without changing the essential robustness constraints.

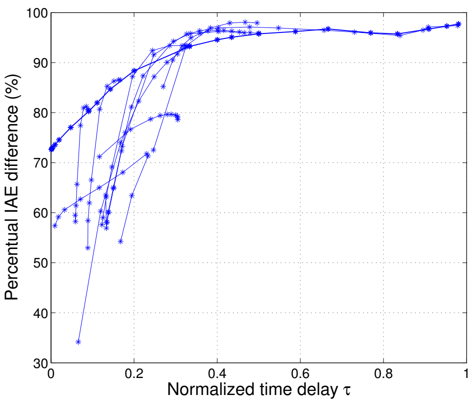

Fig. 6. IAE values comparison for the testbatch using the M-circle constraint alone or both the M_s- and M_p-circles. The plot displays $100 \cdot IAE_{M_s,M_p}/IAE_M$ as a function of τ.

The one subbatch where the newly proposed PIDs gave IAE values with the most deviation from the MIGO ones was

$$P(s) = \frac{1}{(s+1)((sT)^2 + 1.4sT + 1)}, \tag{14}$$
$$\text{with} \quad T = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0.,$$

i.e. processes with complex poles. In particular, when $T = 0.1$, the new IAE is as small as 62.5% of the MIGO IAE, corresponding to a significant improvement. Figure 7 shows the output signal, y, and control signal, u, when a load disturbance, d (see Figure 1), is acting on this process. The dashed curves correspond to the MIGO controller ($K = 3.96$, $T_i = 0.46$, $T_d = 0.08$), the solid lines to the proposed controller with the M-circle constraint ($K = 5.42$, $T_i = 0.29$, $T_d = 0.16$) and the dash-dotted line to the new design method with the M_s- and M_p-constraints ($K = 6.53$, $T_i = 0.22$, $T_d = 0.16$). The open loop Nyquist curves for the three cases are shown in Figure 8. It is known that the MIGO method discards solutions that touches the M-circle twice. This example shows that this choice may be overly conservative. It is also evident that the substitution of the M-circle to the M_s- and M_p-circles gives a much lower IAE-value in this case.

Example 2. (An oscillatory process). Consider an oscillatory system with the linear transfer function

$$P(s) = \frac{9}{(s^2 + s + 9)(s+1)}, \tag{15}$$

which has two poles with a relative damping of $\zeta = 0.17$. An IE-cost function is not suitable for PID design when the system is oscillatory, ruling out use of the MIGO method. The proposed design algorithm, however, can derive a PID design without problems. For $M_s = M_p = 1.4$ the program gave the parameters: $K = 0.37$, $T_i = 0.23$, $T_d = 0.80$. Figure 9 shows the control- and output signals.

6. CONCLUSIONS

This paper has presented a new software tool that can help educate people in PID control systems as well as provide them with controller designs in short time. The controller designs are made to minimize the integrated absolute error during a load disturbance on the process input. The

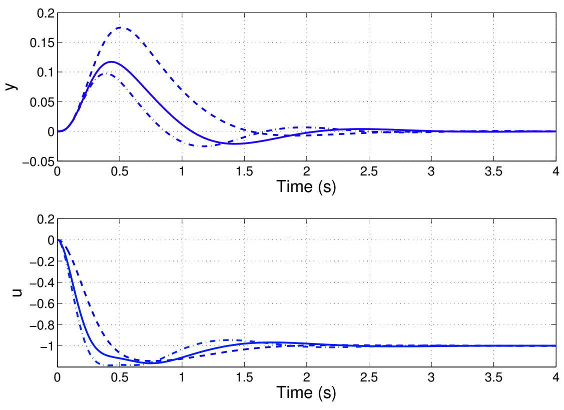

Fig. 7. The output (y) and control signal (u), during a load disturbance, for three different designs on (14), $T = 0.1$. Dashed line: MIGO PID; Solid line: Proposed PID with M-circle constraint; Dash-dotted line: Proposed PID with the M_s- and M_p-circle constraints.

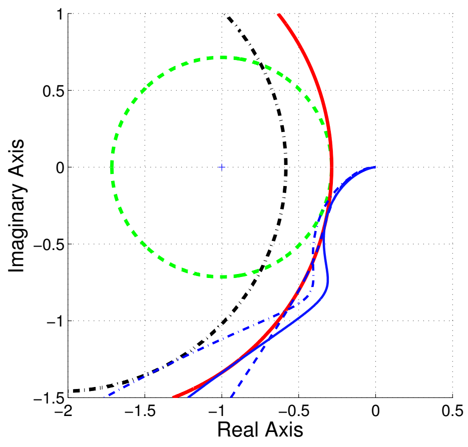

Fig. 8. The open loop Nyquist curves when three different design methods were used on (14), $T = 0.1$. Dashed line: MIGO control; Solid line: Proposed design with M-circle constraint; Dash-dotted line: Proposed design with the M_s- and M_p-circle constraints.

optimization is constrained by robustness conditions on two of the sensitivity functions. The finding that a lot of processes only give one unique minimum solution to the optimization problem lead to the use of the Nelder Mead method. The initial simplex is provided by the AMIGO method, a choice made rather for the speed of the algorithm than it being necessary to find the global minimum.

The software tool was shown to give reasonable controllers on a large batch of processes common in process industry. The use of IAE as cost function also give the possibility to run the program on highly oscillatory systems, as was shown in an example.

Future research should provide a better way of handling the effect of measurement noise on the control signal, providing a sophisticated way of choosing the filter constant T_f. It may also be needed to include even more constraints in the optimization problem in order to, e.g, give robustness to time delay uncertainty. With these modifications it should be possible to use the program for PID controller design on real plants.

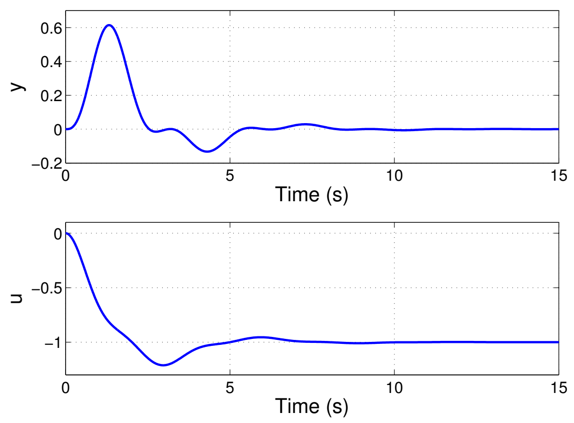

Fig. 9. The output signal, y, and control signal, u, when the proposed design method was used to find a controller for the oscillatory process (15). $M_s = M_p = 1.4$.

ACKNOWLEDGEMENTS

This work was supported by the Swedish Research Council (VR).

REFERENCES

K.J. Åström and T. Hägglund. The future of PID control. *Control Engineering Practice*, 9:1163–1175, 2001.

K.J. Åström and T. Hägglund. *Advanced PID Control*. ISA - The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC 27709, 2005.

T. Hägglund and K.J. Åström. Revisiting the Ziegler-Nichols step response method for PID control. *Journal of Process Control*, 14(6):635–650, 2004.

J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the Nelder-Mead simplex algorithm in low dimensions. *SIAM Journal on Optimization*, 9:112–147, 1998.

Y. Li, K.H. Ang, and G.C.Y. Chong. PID control system analysis and design - Problems, remedies, and future directions. *IEEE Control Systems Magazine*, 26:32–41, 2006.

J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

P. Nordfeldt. PID control of TITO systems. Licentiate Thesis ISRN LUTFD2/TFRT--3238--SE, Department of Automatic Control, Lund University, Sweden, December 2005.

J.J.E. Oviedo, T. Boelen, and P. van Overschee. Robust advanced PID control (RaPID) - PID tuning based on engineering specifications. *IEEE Control Systems Magazine*, 26:15–19, 2006.

H. Panagopoulos, K.J. Åström, and T. Hägglund. Design of PID controllers based on constrained optimisation. *IEE Proceedings - Control Theory & Applications*, 149 (1):32–40, 2002.

F.H. Walters, L.R. Parker Jr, S.L. Morgan, and S. N. Deming. *Sequential Simplex Optimization*. CRC Press LLC, 1991.