IFAC Conference on Advances in PID Control
PID'12
Brescia (Italy), March 28-30, 2012

Robust Semi-automatic Design of PID Controllers for Systems with Time Delay

Guillermo P. Falconí * and **Jürgen E. Ackermann** **

* *Technische Universität München, (e-mail: gpfalconi@mytum.de)*
** *DLR, Oberpfaffenhofen, Germany, (e-mail: ackermann-herrsching@t-online.de)*

Abstract: In this paper an extension of the Matlab-tool PIDrobust is presented. This tool calculates the entire set of PID controllers that stabilizes a set of linear systems with time delay simultaneously. On this basis an iteratively algorithm is used to improve σ-stability in order to assist operators. At the end of the process the operator is able to judge the results and interactively choose a controller. This tool needs much less computational effort than other optimization methods and achieves similar performance. It is can also be used for tuning robust controllers by means of the parameter space approach.

Keywords: PID control, time delay, robust control, linear systems, iterative improvement

1. INTRODUCTION

PID controllers have a wide acceptance in the industry because around 90% of the existing processes can be stabilized and controlled using SISO-PID-Controllers (Koivo and Tanttu (1991)). For instance, in the process industry more than 95% of the controllers use the PID control strategy (Aström and Hägglund (2005)). It has been stated that the popularity of PID controllers relies on the intuitive three terms that characterize their performance (one for the proportional term k_p, one for the integral term k_i and one for the derivative term k_d), on their long history and on the easy introduction of new capabilities like adaption, self-tuning and gain scheduling into PID control (Knospe (2006)).

Nevertheless, the performance of these controllers is often far away of being optimal and the main reason of this problem is the wrong tuning of the controllers (Ender (1993)). The deficiency lies surely not within the lack of tuning rules or methodologies. The popularity of PID control as a research area is rather growing (O'Dwyer (2003)). One of the most used and at the same time one of the oldest methods of tuning PID controllers are the classic laws of thumb by Ziegler and Nichols (Ziegler and Nichols (1942)). For an overview of the tuning methods the reader is referred to Aström and Hägglund (2005); Datta et al. (2000); Tan et al. (1999).

It has been suggested that the reasons of poorly tuned parameters are the lack of knowledge among operators and commissioning personnel, generic tuning methods that do not match with the specific process needs and the large variety of PID structures, which leads to errors during the application of tuning rules (Oviedo et al. (2006)). One of the main problems is the tuning of the derivative term, which in 80% of the cases is switched off or completely omitted (Digest (1996)).

Therefore, the necessity of developing a tool that is easy to operate and with an effective approach for tuning the parameters arises. The Matlab-tool PIDrobust presents a new alternative (Hohenbichler (2009b)). It overcomes some typical limitations of other methods for their applicability in the industry. It can handle any linear plant with time delay and can deal with more than one design criterion. Furthermore, parametric uncertainties can be taken into account.

This method is based on finding the set of all the stabilizing PID controllers (SSC) for a linear plant model. This idea was first presented in Datta et al. (2000) and it was also shown that for a fixed k_p the set of all stabilizing controllers consists of convex polygons in the (k_d,k_i)-plane. In Ackermann and Kaesbauer (2003) the same result was derived using a different approach, the parameter space approach (Ackermann (2002)). The notion of singular frequencies was introduced. Each singular frequency gives rise to a straight line boundary in the (k_d,k_i)-plane, i.e. at a singular frequency a pair of roots crosses the imaginary axis. Subsequently, the theory of finding the SSC has been generalized for systems with time delay (Hohenbichler and Ackermann (2003a,b)).

In this paper, the Matlab-tool PIDrobust has been extended by the idea in Weller and Ackermann (2009), which exposes the possibility of automatic improvement of σ-stability as a support for the synthesis of PID controllers based on the SSC. The σ-stability test can be treated as a stability test using a transformation presented in Ackermann and Kaesbauer (2003). This extension of Matlab-tool PIDrobust is thought as a support for the operators for tuning the PID controllers. At the end of the process the operator should be able to judge the results, as a better σ-value not always yields to a better performance of the controller. Therefore, the evaluation of the controller is done in the preferred representation of the operator using the LTIviewer of Matlab, e.g. impulse and step responses, zero-pole-map, Nyquist or Bode plots.

The remainder of this paper is organized as follows. In the next section, the problem is stated. In Section 3 the calculation of the SSC is recapitulated. The algorithm to iteratively improve σ-stability is presented in Section 4. In Section 5 some examples are presented to show how the Matlab-tool works and to demonstrate its performance. Finally, in Section 6 the results of this paper are summarized.

2. PROBLEM STATEMENT

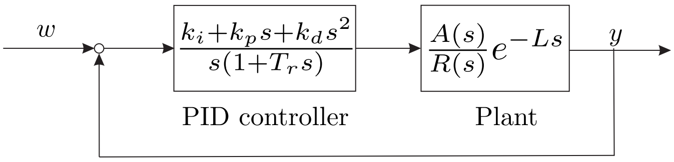

Fig. 1. System structure

In this paper, the system seen in Figure 1 is considered. It consists of a PID controller

$$PID\,(s) = \frac{k_d s^2 + k_p s + k_i}{s\,(1+T_r s)} \tag{1}$$

and a linear plant with time delay and unknown but constant parameters $\mathbf{q}$

$$G\,(s,\mathbf{q}) = \frac{A(s,\mathbf{q})}{R(s,\mathbf{q})}e^{-Ls}. \tag{2}$$

The characteristic function of the closed-loop is

$$P(s,\mathbf{q}) = (k_i + k_p s + k_d s^2)A(s,\mathbf{q}) + \underbrace{B(s,\mathbf{q})e^{Ls}}_{\hat{B}(s,\mathbf{q})}, \tag{3}$$

where $L \geq 0$ is the time delay and $A(s,\mathbf{q})$ and $B(s,\mathbf{q})$ are polynomials with real coefficients

$$\begin{aligned} A(s,\mathbf{q}) &: = a_0(\mathbf{q}) + a_1(\mathbf{q})s + ... + a_m(\mathbf{q})s^m,\; a_m \neq 0, \\ B(s,\mathbf{q}) &: = s\,(1+T_r s)\,R(s,\mathbf{q}) \\ &= b_0(\mathbf{q}) + b_1(\mathbf{q})s + ... + b_n(\mathbf{q})s^n, \quad b_n \neq 0. \end{aligned} \tag{4}$$

The uncertain parameters, including the time delay, are specified in an operating domain

$$Q = \left\{ (L,\mathbf{q})^T \,|L \in \left[L^-;L^+\right], q_i \in \left[q_i^-;q_i^+\right] \right\}, \tag{5}$$

where $(\cdot)^-$ and $(\cdot)^+$ mean lower and upper bounds respectively. To take the uncertain parameters into account the parameter space approach is used. The approach consists of two steps. The first one is the controller synthesis step, where the set of controllers that stabilize a finite number of plant representatives, e.g. the vertices of Q, is searched. The second step is the robustness verification. For a given controller the stability region in the plant's parameter space should enclose Q entirely. In this paper we will focus on the controller synthesis and thus consider $(L,\mathbf{q})^T$ as fixed values. For the analysis of the second step the reader is referred to Hohenbichler and Abel (2006); Ackermann (2002); Silva et al. (2005).

3. SET OF ALL STABILIZING PID CONTROLLERS

In the following, the calculation of the SSC in the controller parameter space $(k_d, k_i, k_p)^T$ is explained. It is based on finding the stability boundaries in the $(k_d$-$k_i)$-plane for a fixed k_p and on finding the stability-enabling k_p intervals. This approach has the advantage that this intervals can be calculated using the function $k_p = f(\omega)$, see (6). Furthermore, in the $(k_d$-$k_i)$-plane stability boundaries are straight lines and thus computationally favorable. In this paper, the first step will be summarized and for details on the second step the reader is referred to Hohenbichler (2009a). In the next section, the Hurwitz stability case is treated and subsequently the σ-stability case.

3.1 Hurwitz stability

A property of the considered system with time delay is that (3) is a quasipolynomial and thus has an infinite number of roots. The closed loop is Hurwitz stable if all the roots lie in the open left half plane (LHP) and large roots do not approach the imaginary axis asymptotically (Bellman and Cooke (1963)). Furthermore, it is required that $l := n - m \geq 2$ because of the *principal term condition* (Pontryagin (1955)). This is assumed in the following. In order to calculate stability regions in the parameter space, the root boundaries are searched, i.e. where a root crosses the imaginary axis. There exist three types of such boundaries:

- real root boundaries (RRB) $s = 0$,
- complex root boundaries (CRB) $s = \pm j\omega$,
- and infinite root boundaries (IRB) $s = \infty$.

For the following description of the boundaries, the auxiliary functions

$$f(\omega) := f_1(\omega)sin(\omega L) + f_2(\omega)cos(\omega L) \tag{6}$$

$$g(\omega) := \omega\,(-f_2(\omega)sin(\omega L) + f_1(\omega)cos(\omega L)) \tag{7}$$

are defined, where

$$f_1(\omega) := \frac{-R_A R_B - I_A I_B}{\omega\,(R_A^2 + I_A^2)},\; f_2(\omega) := \frac{I_A R_B - R_A I_B}{\omega\,(R_A^2 + I_A^2)}. \tag{8}$$

R and I denote the real and imaginary parts of the polynomials. For further details on the derivation see Hohenbichler (2009b).

The RRB is found by inserting $s = 0$ in (3)

$$P(0) = k_i A(0) + B(0) = 0 \tag{9}$$

which leads to

$$k_i = -\frac{b_0}{a_0}. \tag{10}$$

Thus, a RRB exists for $a_0 \neq 0$.

For finding the CRBs the characteristic polynomial $P(s)$ is divided in real and imaginary parts. Hence, the requirement for a root at $s = j\omega$ can be expressed

$$\underbrace{\begin{pmatrix} R_A & -\omega^2 R_A \\ I_A & -\omega^2 I_A \end{pmatrix}}_{S_f} \begin{pmatrix} k_i \\ k_d \end{pmatrix} + \begin{pmatrix} R_{\hat{B}} - k_p\omega I_A \\ I_{\hat{B}} + k_p\omega R_A \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{11}$$

in matrix form. Only positive values of ω are taken into account as the roots are always complex conjugates. For fixed k_p and ω the matrix S_f is singular. Thus, the linear equation system (11) can only be solved if

$$det \begin{pmatrix} R_A & R_{\hat{B}} - k_p\omega I_A \\ I_A & I_{\hat{B}} + k_p\omega R_A \end{pmatrix} = 0. \tag{12}$$

This equation can be brought to the form

$$k_p = f(\omega), \tag{13}$$

whose positive solutions are the *singular frequencies* ω_η. The set of all singular frequencies is denoted as Ω. Subsequently, inserting (13) in (11) leads to

$$k_i = \omega_\eta^2 k_d + g(\omega_\eta) \tag{14}$$

a straight line, the CRB, with positive slope in the (k_d,k_i)-plane for each singular frequency.

In the case of a delay-free system, the number of CRBs is finite and thus the stability region consists of a set of convex polygons. Furthermore, the characteristic polynomial $P(s)$ has a root at $|s| = \infty$ if its largest coefficient

$$\begin{array}{r} b_n \text{ , for } l > 2, \\ b_n + a_m k_d \text{ , for } l = 2 \end{array} \tag{15}$$

equals 0. As $b_n \neq 0$ and $a_m \neq 0$, this happens only for $l = 2$. The IRB is then

$$k_d = -\frac{b_n}{a_m}. \tag{16}$$

For analyzing the systems with time delay, an infinite number of CRBs must be managed (see (13)) and therefore the concept of root chains is used. Root chains describe the asymptotic behavior of large roots. Hence, the set of large singular frequencies is defined as

$$\Omega_{\omega_l} := \{\omega_\eta \in \Omega \mid \omega_\eta > \omega_l\}, \tag{17}$$

which depends on a chosen fix ω_l.

In Hohenbichler (2009b) it is shown that an ω_l exists so that, for large singular frequencies, the absolute value of the intersection points of the CRBs of $\omega_\eta \in \Omega_{\omega_l}$ with the k_d-axis approaches ∞, for $l > 2$, and to $\left|\frac{b_n}{a_m}\right|$, for $l = 2$, strictly monotonically.

Because of this kinematic of the CRBs and the fact that their slope increases with ω_η^2 as shown in (14), it can be proved that a singular frequency ω_l exists so that the CRBs corresponding to larger frequencies do not change the stable region in the case of a retarded system with time delay ($L > 0$, $l > 2$) (Hohenbichler (2009b)). Thus, if a stable region exits it is composed of a set of convex polygons.

In the case of neutral systems, i.e. $l = 2$, it is possible that the CRBs of large singular frequencies do not affect the stable region but this is not always true. In these cases the stable region is the limit of a sequence of convex polygons and can be approximated conservatively by a set of convex polygons. The approximation error can be approximated and reduced arbitrarily by taking CRBs of higher frequencies into account (Hohenbichler (2009a)).

Furthermore, in the case $l = 2$ IRBs exist. For large s the characteristic polynomial $P(s)$ can be approximated by

$$\begin{aligned} \widetilde{P}(s) &= k_d a_m s^n + b_n s^n e^{sL} \\ &= s^n \left(k_d a_m + b_n e^{sL}\right), \end{aligned} \tag{18}$$

and the real part of the roots

$$\sigma_\infty = \frac{1}{L} ln \left(\left|\frac{k_d a_m}{b_n}\right|\right). \tag{19}$$

can be found. Therefore, if a stable region exists it lies within the lines

$$|k_d| < \left|\frac{b_n}{a_m}\right|. \tag{20}$$

As these boundaries represent the crossing of an infinite number of roots at the imaginary axis, no stable region exists outside this lines.

In this section it was shown that the RRB (10), the CRBs (14) and the IRBs (16) and (20) are straight lines in the $(k_d$-$k_i)$-plane. Thus, the stable region is formed by a set of convex polygons or can be approximated by one.

3.2 σ-stability

σ-stability means that all the poles of the closed-loop lie left from a parallel to the imaginary axis straight line. This line is located at the chosen σ_0 value. In Ackermann and Kaesbauer (2003) the transformation

$$s := v + \sigma_0 \tag{21}$$

is given, which maps the straight line at σ_0 in the s-plane to the imaginary axis of the v-plane. Thus, by applying this transformation to the characteristic function (3)

$$\begin{aligned} P'(v) = &\left[\left(k_i + k_p\sigma_0 + k_d\sigma_0^2\right) + \left(k_p + 2k_d\sigma_0\right)v + k_d v^2\right] \cdot \\ &\cdot A\left(v + \sigma_0\right) + \widehat{B}\left(v + \sigma_0\right) \end{aligned}$$

the σ-stability of $P(s)$ can be treated as the stability of $P'(v)$. Furthermore, as $P'(v)$ can be brought to an equivalent structure of (3) through

$$\begin{aligned} k_i' &:= k_i + k_p\sigma_0 + k_d\sigma_0^2 \text{ ,} \\ k_p' &:= k_p + 2k_d\sigma_0 \text{ ,} \\ k_d' &:= k_d \text{ ,} \\ A'(v) &:= A(v + \sigma_0) \text{ ,} \\ \widehat{B}'(v) &:= \widehat{B}(v + \sigma_0) \text{ ,} \end{aligned} \tag{22}$$

the computation of the set of all σ-stabilizing PID controllers can be analogously treated with the methods presented in Section 3.1.

4. ITERATIVE σ - IMPROVEMENT

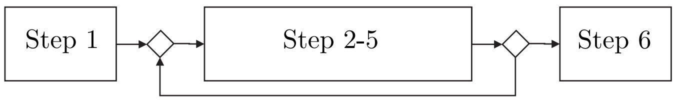

Fig. 2. Automatic iterative σ - improvement

The control design process proposed in this paper consist of an automatic iterative σ-improvement and an analysis done by the operator about the effects of this improvement, e.g. the impact on other design criteria. A scheme of the automatic algorithm, based on Weller and Ackermann (2009), can be seen in Figure 2:

(1) Initialization: the initial value of $\sigma_0 = 0$ is set. This is equivalent to Hurwitz stability. Furthermore, the parameters σ-tolerance and the maximum number of iterations are also set. These parameters are used to control the break of the loop.
(2) Calculate the σ-stable region: first the k_p-grid is computed. Subsequently, for each k_p and each plant the stable region in the (k_d,k_i)-plane is computed. The stable regions are composed of convex polygons.
(3) Calculate the polygons' intersection: for each k_p the intersection of the stable regions of all the plants is computed. The intersection is also composed of convex polygons.

(4) Calculate centers of gravity: the centers of gravity of each polygon are computed. The center of gravity of each polygon is the most robust point in the sense, that it is the most distant point in the polygon to the stability boundaries.
(5) Calculate the new σ_0 value: the σ_0 value is computed for every center of gravity, which corresponds to the worst σ_0 value of all plants. This guarantees that the real part of the roots of all plants lie left from σ_0. Subsequently, the best σ_0^* value of all centers of gravity is selected as the basis for the next iteration step, i.e. the most negative one.
(6) Controller selection: the user selects a controller, e.g. the center of gravity corresponding the σ_0^* value at the last iteration step. Subsequently, the closed-loop can be analyzed with e.g. Nyquist plots, step responses, impulse responses, Bode plots or zero-pole-maps. The tool provides also the possibility of comparing the σ-best PID controllers of each iteration.

The iteration loop breaks if the maximum number of iterations is exceeded or if σ-tolerance $> \left|\sigma_{0,i}^* - \sigma_{0,i-1}^*\right|$.

5. EXAMPLES

In this section two examples are presented in order to show the performance of the PIDrobust program. The following example is a linear plant with time delay

$$G_2(s) = \frac{1}{(1+0.2s)^2} e^{-s} \tag{23}$$

presented in Aström and Hägglund (2000) as one of the 20 benchmark plants. For the first five iterations of the σ-iteration process, the parameters of the best controllers are given in Table 1, assuming $T_r = 0$. Furthermore, the parameters of a controller tuned by the software package PIDEasy are listed. PIDEasy uses automatic simulations to search globally for controllers that meet five given design objectives (Li et al. (2006)).

Table 1. Example 1 - Controllers' parameters

		PID Parameters			Resultant Margins	
Iteration	σ_0	k_p	k_i	k_d	GM	PM(°)
1	-0.926	0.552	0.890	0.133	2.052	50.470
2	-1.377	0.467	0.688	0.102	2.453	60.942
3	-1.626	0.404	0.616	0.077	2.700	63.082
4	-1.786	0.368	0.576	0.064	2.880	64.128
5	-1.884	0.337	0.549	0.056	3.015	64.827
PIDEasy	-1.350	0.3	0.509	0.051	3.311	65.8

Figure 3 shows the step response of the closed-loop using the controllers found by the σ-iteration algorithm. It can be seen that the step response is improved every iteration, i.e. with a better σ_0 value. In Figure 4, the step response of the closed-loop is plotted using the last controller of the σ-improvement algorithm and the one tuned by PIDEasy. It can be seen that the step responses are similar, but the one found by the σ-improvement algorithm has a better settling time. For instance, depending on the overshoot's restrictions, the controller found in the third iteration could be an alternative.

In the last example, it can be seen that the performance of the PID controller that can be achieved with the σ-improvement algorithm is similar to other tuning methods. Most of the methods for tuning PID controllers are

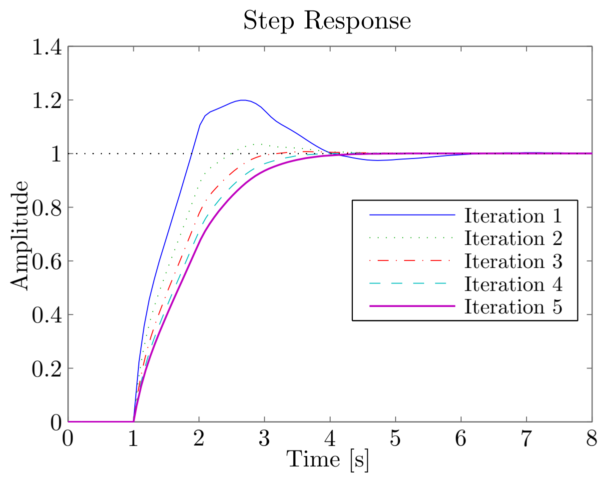

Fig. 3. Example 1 - step response

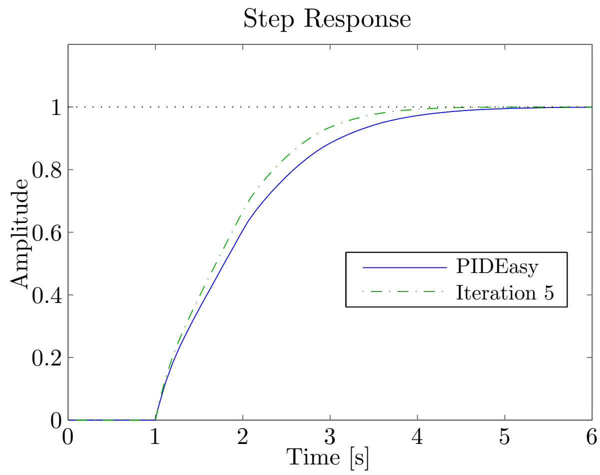

Fig. 4. Example 1 - step responses' comparison

based on maximizing or minimizing some properties of the closed-loop for one plant. The method presented in this paper distinguishes from the other methods because it is capable to handle this process for more than one plant using the same controller.

This allows to tune a controller that is robust against parameter uncertainty using the parameter space approach. For example, the following plant

$$G_3(s) = \frac{-0.5s^4 - 7s^3 - 2s + 1}{s^6 + 11s^5 + q_1 s^4 + 95s^3 + 109s^2 + 74s + 24} e^{-0.08s}$$

is given in Hohenbichler (2009b) with the nominal value $q_1 = 46$. It is assumed that the uncertain parameter q_1 may vary within the following operating domain $q_1 \in [40, 50]$. It is searched for a controller that stabilizes the plant robustly. Therefore, we calculate the set of controllers that can stabilize the two plants representing the vertices of the operating domain, which is in this case one-dimensional. In Figure 5, the stable region and the centers of gravity are shown in the parameter space for these two plants. Plant 1 corresponds to $q_1 = 40$ and plant 2 corresponds to $q_1 = 50$. The intersection of the two regions represents the set of controllers that can stabilize the two plants simultaneously.

In Table 2 the resulting controllers are listed along with the corresponding gain and phase margins for every plant.

Table 2. Example 2 - controllers' parameters

		PID Parameters		
Iter.	σ_0	k_p	k_i	k_d
1	-0.101	2.724	3.773	-0.336
2	-0.182	3.252	3.386	1.056
3	-0.230	3.707	2.925	1.449
4	-0.256	3.766	2.709	1.561
5	-0.271	3.749	2.595	1.603
		Resultant Margins		
	GM1	PM1(°)	GM2	PM2(°)
1	1.737	50.25	1.425	50.27
2	1.893	56.41	1.658	56.42
3	1.548	63.00	1.603	63.01
4	1.458	65.74	1.599	65.74
5	1.432	67.08	1.611	67.08

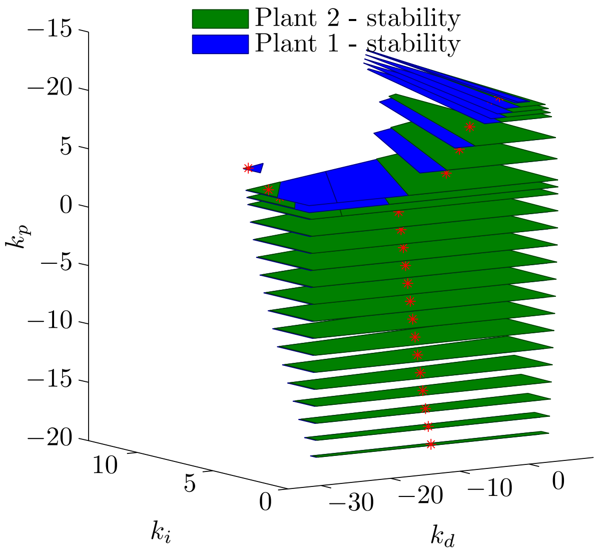

Fig. 5. Example 2 - PID parameter space

The σ_0 value corresponds to the plant with the worst σ_0 value that is achieved with the controller. It can be seen that improving the σ_0 value does not always leads to an improvement of the gain or phase margins.

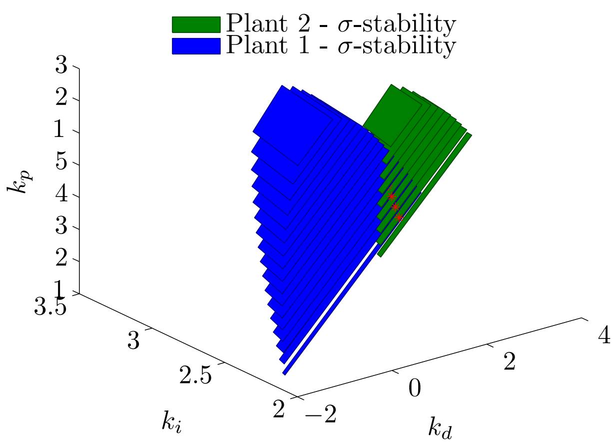

Fig. 6. Example 2 - PID parameter space, $\sigma_0 = -0.25642$

In Figure 6 the σ-stable region of the fifth iteration is shown in the parameter space for the two plants. In Figures 7 and 8 the variation of the step responses during the optimization process can be seen for every plant. As long as the intersection of the σ-stable regions is not empty, there exists a controller that σ-stabilizes the two plants simultaneously. It is denoted that the procedure used in this example for stabilizing two plants simultaneously can be extended without any limitations to a higher number of plants.

Having explored examples that show the potential of the tuning method presented in this work, the next section summarizes the paper and gives a perspective of future work based on the presented results.

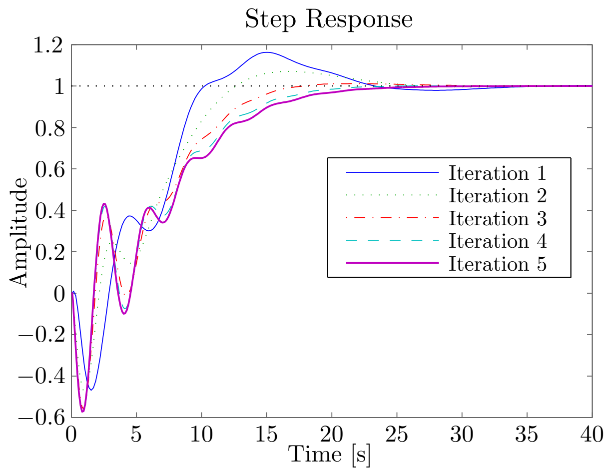

Fig. 7. Example 2 - step responses of plant 1

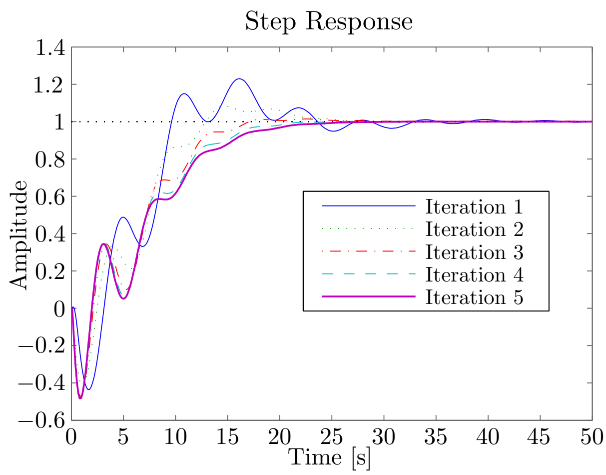

Fig. 8. Example 2 - step responses of plant 2

6. CONCLUSION

In this paper, the approach of a robust semi-automatic design for PID controllers has been presented. This approach is based on calculating the set of all stabilizing PID controllers in the parameter space for a given linear plant with time delay. This is done by gridding k_p and calculating the stable convex polygons in the (k_d,k_i)-plane.

It is a robust design in so far as the approach can find controllers that stabilize more than one plant simultaneously. In this way parameter uncertainties can be handled. This is done by intersecting the stable polygons of all the plants in order to find the set of all PID controllers that stabilizes all the plants. This set is then the basis for the σ-improvement algorithm.

The σ-stability case can be treated analogously to the stability case using the mapping given in (21) and the parameter transformation given in (22). During the σ-improvement algorithm first the centers of gravity of all polygons are calculated. Subsequently, the corresponding σ_0 values are computed and the best one is the basis for the next iteration step. Then the process is carried on by calculating the σ_0-stable region.

This approach is semi-automatic because after the σ-improvement process the operator is able to judge the resulting closed-loop and choose the best suitable controller. For this purpose, the Matlab's analyzing tool LTI-viewer is used. This is convenient as the performance of the closed-loop is not only characterized by the σ-stability but other requirements should also be taken into consideration. This evaluation can be performed faster and easier if it is done by the operator, e.g. using graphical information like the step response of the closed-loop. In the Matlab-tool PIDrobust this is implemented interactively.

Future work can be made using the same principle of the σ-improvement algorithm for improving other criteria. Instead of calculating the σ_0 value of the centers of gravity, the gain and phase margins could be used. Other possibilities are the circle-stability for delay-free systems and the large roots location for systems with time delay as presented in Hohenbichler (2009b). Furthermore, a multi-criteria improvement process can be achieved. This cannot be done using only a polygon intersection because the (k_d,k_i)-planes of all the criteria may not be parallel, but a polyhedra intersection function could be used instead.

Further research can be made in analyzing the improvement potential of the optimization process for unbounded regions. Currently, the algorithm calculates the centers of gravity of the polygons that are plotted in case that unbounded regions are found. Hence, the defined bounding box used for the plot has an influence in the centers of gravity and thus in the improvement algorithm.

The method presented in this paper has some advantages compared to the plenty of methods available nowadays. One of these advantages is that this method is able to handle more than one plant at the same time. Another advantage is that the kind of plants that can be analyzed is not restricted to a limited group but all linear plants with and without time delay can be treated. Moreover, in this work the robust semi-automatic design for PID controllers has been implemented in such a way that the user can use the tool intuitively without understanding the underlying mathematical basis. Finally, the computational effort is less than a global optimization as the stability boundaries are directly computed. The presented extension of PIDrobust is published at *www.irt.rwth-aachen.de/en/fuer-studierende/downloads/pidrobust/*.

REFERENCES

Ackermann, J. (2002). *Robust Control: The Parameter Space Approach.* Springer.

Ackermann, J. and Kaesbauer, D. (2003). Stable polyhedra in parameter space. *Automatica*, 39, 937–943.

Aström, K.J. and Hägglund, T. (2000). Benchmark systems for PID control. *Proc. IFAC Workshop, Terrasa, Spain*, 165–166.

Aström, K.J. and Hägglund, T. (2005). *Advanced PID Control.* Research Triangle Park, NC: ISA - The Instrumentation, Systems, and Automation Society.

Bellman, R. and Cooke, K.L. (1963). *Differential-Difference Equations.* Academic Press.

Datta, A., Ho, M.T., and Bhattacharyya, S.P. (2000). *Advances in Industrial Control - Structure and Synthesis of PID Controllers.* Springer.

Digest, I. (1996). Getting the best out of PID machine control. *Digest IEE PG16 Colloquium.*

Ender, D.B. (1993). Process Control Performance: Not as Good as you Think. *Control Engineering*, 40, 180–190.

Hohenbichler, N. (2009a). All stabilizing PID controllers for time delay systems. *Automatica*, 45(11), 2678 – 2684. doi:10.1016/j.automatica.2009.07.026.

Hohenbichler, N. (2009b). *Robuster Reglerentwurf auf Grundlage der Menge aller stabilisierenden PID-Regler.* Ph.D. thesis, RWTH Aachen.

Hohenbichler, N. and Abel, D. (2006). Robust pid-controller design meeting pole location and gain/phase margin requirements for time delay systems. *at-Automatisierungstechnik*, 54(10), 495–501. doi:doi:10.1524/auto.2006.54.10.495.

Hohenbichler, N. and Ackermann, J. (2003a). Computing Stable Regions in Parameter Spaces for a Class of Quasipolynomials. *IFAC Workshop on Time Delay Systems, Rocquencourt.*

Hohenbichler, N. and Ackermann, J. (2003b). Synthesis of Robust PID Controllers for Time Delay Systems. *Proceedings of European Control Conference, Cambridge.*

Knospe, C. (2006). PID Control. *IEEE Control Systems Magazine*, 26, 30–31.

Koivo, H. and Tanttu, J. (1991). Tuning of PID Controllers: Survey of SISO and MIMO techniques. *Proceedings of the IFAC Intelligent Tuning and Adaptive Control Symposium*, 75–80.

Li, Y., Ang, K.H., and Chong, G.C. (2006). PID Control System Analysis and Design - Problems, remedies and future directions. *IEEE Control Systems Magazine*, 26, 32–41.

O'Dwyer, A. (2003). PID compensation of time delayed processes 1998-2002: a survey. *Proceedings of the American Control Conference*, 1494–1499.

Oviedo, J.J.E., Boelen, T., and Overschee, P.V. (2006). Robust Advanced PID Control (RaPID) - PID Tuning Based on Engeneering Specifications. *IEEE Control Systems Magazine*, 26, 15–19.

Pontryagin, L.S. (1955). On the zeros of some elementary trascendental functions. *American Mathematical Society Translation*, 2, 95–110.

Silva, G.J., Datta, A., and Bhattacharyya, S. (2005). *PID Controllers for Time-Delay Systems.* Birkhäuser Boston.

Tan, K., Wang, Q., Hang, C., and Hagglund, T. (1999). *Advances in PID Control.* Springer.

Weller, R. and Ackermann, J. (2009). Automated Design for Robust PID Controllers. *Proceedings of the 6th IFAC Symposium on Robust Control Design, Haifa*, 19.

Ziegler, J.G. and Nichols, N.B. (1942). Optimum settings for automatic controllers. *ASME Transactions*, 64, 759–768.