Rule-based PI Controller Autotuning for Drive Systems

Martin Pfeifer, Thomas Weickert, Silke Klose

***Abstract*— Tuning a PI controller can be quite cumbersome for the non-expert as the closed-loop control system has to meet various requirements while the influence and interaction of the two degrees of freedom are not always clear. This paper addresses the design of an iterative PI controller autotuning for drive systems with the idea of imitating a human expert. In contrast to existing concepts, a new approach with multiple tuning strategies is applied which gives a compact rule base that is easy to modify. The performance of the proposed algorithm is illustrated through motion control testbench trials.**

I. Introduction

PI controllers are the dominating control structure for a wide range of technical processes. The performance of these systems is highly dependent on the respective controller setting. Once a controller is configured it provides satisfying results for the process, given an unchanged operating environment. However, if process variables change, the control performance suffers. Hence, modern PI controllers come with autotuning functionalities which enable an automatic adaption of the controller to its application.

Most existing autotuning methods are identification-based which means that a certain model structure is assumed and system parameters are identified. Based on the estimated system parameters the controller parameters are calculated. Such methods can only work well if the model structure fits to the system at hand. Nonlinearities, as e.g. limitations of drives or friction, are usually neglected by the model, which can lead to unspecified control behavior. Thus, tuning a system in closed-loop operation is meaningful, since the real system behavior is considered.

The basis for rule-based autotuning is the advance of an experienced control engineer configuring a controller. The human expert observes the response of the system to an excitation signal and, based on characteristics of the response, changes the controller parameters according to certain rules which were acquired by experience. This process is repeated until no improvement can be attained anymore. The idea of rule-based selftuning is to imitate the human expert by means of a fuzzy logic rule base which contains the available expertise. The design of a suitable fuzzy system is subject of many publications, as [1]–[8]. The proposed concepts differ w.r.t. the applied excitation signal, system response characterization and rule base. Fig. 1 illustrates the basic structure of the rule-based selftuning approach. The involved variables are the set point $w(t)$, the manipulated variable $u(t)$, the controlled variable $y(t)$, the number of system response characteristics p and the tuning factors f_{kr} and f_{Tn}.

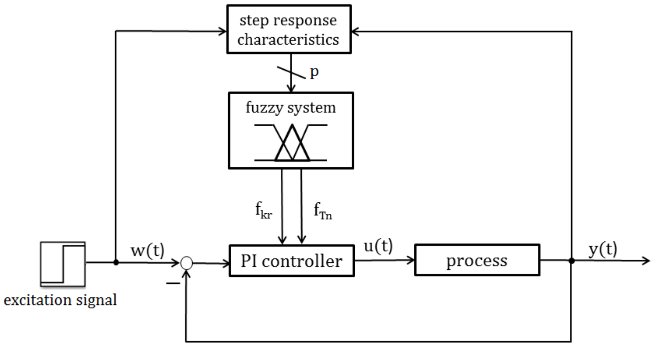


Fig. 1. Basic structure of rule-based autotuning

All of the above-mentioned rule-based autotuning procedures are designed with a single rule base. Hence, for treating various adverse controller settings, the system response has to be characterized in a complex way yielding an extensive rule base. For these concepts, rule base modifications, e.g. for considering process-specific nonlinearities, are hardly possible. Moreover, most approaches do not consider the presence of measurement noise. Thus, there is need for further research taking these constraints into account.

The paper is organized as follows. Initially, in Section II the problem is defined. Existing concepts are investigated and the need for an additional approach is motivated. In Section III the design of a new algorithm is delineated. The basic idea is presented and the rule base is extensively derivated. Moreover, a new approach for the characterization of non-ideal system responses is described. Section IV illustrates the proposed autotuning algorithm by means of a particular motion control problem. Finally, Section V concludes with a brief summary of the paper.

II. Problem Definition and Requirements

Consider the following mathematical representation of a PI controller

$$G_{\text{PI}}(s) = k_r \left(1 + \frac{1}{sT_n}\right), \tag{1}$$

M. Pfeifer is with the Department of Electrical Engineering and Information Technology, Institute of Control Systems, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany. martin.pfeifer@kit.edu
Thomas Weickert and Silke Klose are with ABB AG, Corporate Research Center Germany, 68526 Ladenburg, Germany. {thomas.weickert,silke.klose}@de.abb.com

where k_r and T_n are the proportional gain and the integral time constant, respectively. The objective is to find suitable values for k_r and T_n such that a desired control behavior is attained. For this purpose, the system is excited by a desired reference signal and the system response is characterized by a number of parameters. In some existing approaches the excitation signal is given by a reference step with the step response being characterized by e.g. the overshoot (OS), the undershoot (US) and the settling time [4], [5]. A rule base evaluated by fuzzy logic is used for mapping these parameters to suitable tuning factors f_{kr} and f_{Tn}. Using these tuning factors, the controller parameters are tuned by

$$k_{r,i+1} = f_{kr,i} \cdot k_{r,i}, \tag{2a}$$

$$T_{n,i+1} = f_{Tn,i} \cdot T_{n,i}, \tag{2b}$$

for $i = 0, ..., k-1$. The procedure runs iteratively until a stopping criterion is fulfilled. The complexity of the rule base is strongly dependent on the number of system response characteristics p serving as input variables for the fuzzy system. If each of the p linguistic input variables has q_j linguistic values ($j = 1, ..., p$), a complete and unambiguous rule base would consist of the following number of rules

$$N_{\text{rules}} = \prod_{j=1}^{p} q_j. \tag{3}$$

In preparing this study, a detailled theoretical and practical analysis of the methods proposed in [4] and [5] was accomplished in the context of a motion control application. At this, the algorithm proposed in [4] produced unsatisfactory results. The algorithm proposed in [5] provided partially good results, however, accompanied by several problems for certain initial controller parameters. A modification of the algorithm, in order to consider process-specific pecularities and non-idealities, proves to be difficult, since the rule base consists of 184 rules arising from three input variables (*OS*, *US*, *status*). Many of the rules are not motivated in this publication and there still exist gaps in the proposed rule base leading to unspecified behavior. Moreover, there is no concept contained for dealing with non-ideal measurements.

Due to the reasons mentioned above it seemed meaningful to create a new algorithm taking all the drawbacks of existing concepts into account. We aim at a compact, parsimonious, transparent and complete rule base which can be easily modified in order to incorporate process-specific requirements. The new algorithm should enable a fast convergence to the performance maximum in the controller parameter space for all initial controller parameters. Moreover, a robust determination of the system response characteristics which is insensitive to measurement noise is aspired.

III. Design of the New Algorithm

A. Basic Idea

The first interim objective is the design of a parsimonious fuzzy system with a compact rule base. As illustrated in (3), the number of rules is significantly dependent on the number of system response characteristics p. Thus, in this paper, the closed-loop step response is characterized by only a single parameter - the damping ratio (or just "damping") d. The idea is inspired by the approximation of processes with second order systems of

$$G(s) = \frac{K}{1 + 2dTs + T^2 s^2}. \tag{4}$$

Regarding this approximation, there is an unambiguous mathematic relation between the damping d and the first OS and US of the step response relative to the set point

$$\text{OS} = \exp\left(-\frac{\pi d}{\sqrt{1-d^2}}\right), \quad \text{US} = \exp\left(-\frac{2\pi d}{\sqrt{1-d^2}}\right). \tag{5}$$

Consequently, for processes that can be approximated by a second order system, the characterization by the damping is more effective compared to the characterization by the OS and US. Regarding drive systems, this approach is particularly suitable for one-mass systems as the closed-loop is then described by a second order system. Since the damping provides a high-level characterization of the closed-loop behavior, the approach often proves appropriate, even for processes that cannot be approximated by second order systems.

A major problem of rule-based autotuning algorithms is that similar insufficient system responses can be ascribed to multiple adverse parametrizations of the PI controller. Most of the existing concepts try to solve this problem by a detailled characterization of the system responses. In this paper we apply an alternative, very human-like approach: for each input d to the rule base, there exist two tuning strategies. Each strategy is applied as long as it is successful and the tuning strategy is switched if a tuning iteration is not successful. The algorithm terminates if no strategy is able to attain control improvement anymore.

An essential question in this context is how *success* can be evaluated. OS, US or achieved damping could be possible quantificators for this purpose [4], [5]. However, the objective of a control system is not to create a certain OS and US but to optimize reference tracking and disturbance rejection with respect to a meaningful controller action. Inspired by the optimal control approach a performance index J is specified:

$$J = \frac{1}{N} \sum_{k=1}^{N} \left(|e(k)| + r \cdot |u(k)|\right). \tag{6}$$

In this equation N represents the number of samples during a step response recording. The first part of the sum penalizes a bad control performance by adding up values of the absolute control error $e(k) = w(k) - y(k)$. The second part of the sum penalizes controller action $u(k)$ which can be weighted by the factor $r \in \mathbb{R}_{\geq 0}$.

As for most existing rule-based autotuning concepts, the excitation signal is chosen to be a reference step with the corresponding closed-loop step response as system response. In order to take disturbance rejection into account, an

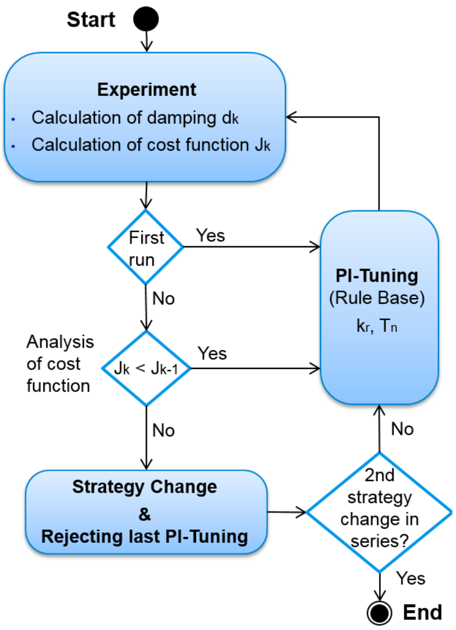


Fig. 2. Flow chart of the proposed method

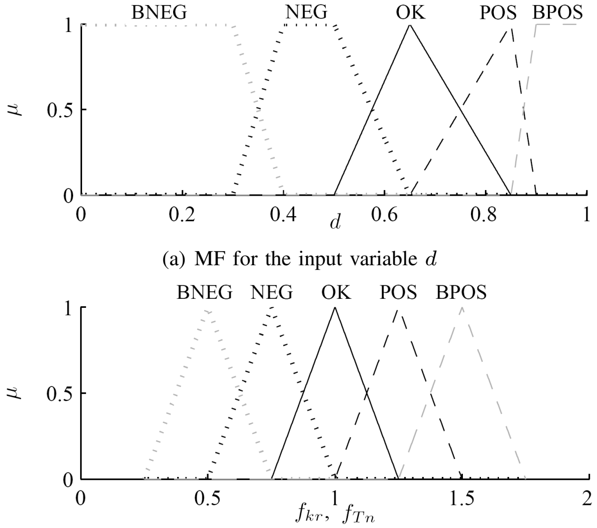
additional disturbance step is applied to the system after the step response has sufficiently settled. A tuning step is then rated as successful if the performance index J of the system response has decreased, compared to its value in the previous iteration. Considering two strategies yields then the algorithm depicted in the program flow chart in Fig. 2. Without much effort the algorithm can be extended by adding further strategies which gives a great flexibility.

B. Rule Base

Based on the objectives presented at the end of section II, the following requirements are defined:

- The membership functions have to be designed wide enough to ensure robustness w.r.t. irregularities.
- The membership functions should be finely partitioned to enable a fast tuning procedure.
- A damping around 0.7 is aspired. Regarding a second order system, 0.7 corresponds to a step response OS of 4.4% which nearly represents the tuning goal in [5].
- The rule base has to cover difficult dynamical exceptions, e.g. the case where an increasing proportional gain leads to a decreased OS in the step response.
- At least one tuning factor should always be significantly far away from 1. Otherwise a small modification in both controller parameters can lead to very small changes in the step response leading to even smaller controller parameter variations in the next iterations.
- The optimization of k_r and T_n should be decoupled if possible. This enables free progression in the controller parameter space and finally a parameter configuration which is close to a possible minimum of (6).

The membership functions (MF) for the linguistic input and output variables are shown in Fig. 3 (a) and (b), respectively. The values of the linguistic variables are BNEG (big negative), NEG (negative), OK, POS (positive), and BPOS (big positive). The damping is said to be "OK" between 0.5 and 0.85 which, in a second order system, corresponds to an OS from 0.6% to 16%. If the damping is too high or too low, the linguistic variables distinguish between two intensities each, which allows a faster convergence.



(a) MF for the input variable d

(b) MF for output variables f_{kr}, f_{Tn}

Fig. 3. MF for the input and output variables of the fuzzy system

$d =$	BNEG	NEG	OK	POS	BPOS
ver 1	$k_r \downarrow\downarrow$	$k_r \downarrow$	$k_r \rightarrow$	$k_r \uparrow$	$k_r \uparrow\uparrow$
	$T_n \uparrow\uparrow$	$T_n \uparrow$	$T_n \rightarrow$	$T_n \downarrow$	$T_n \downarrow\downarrow$
ver 2	$k_r \downarrow\downarrow$	$k_r \downarrow$	$k_r \rightarrow$	$k_r \uparrow$	$k_r \uparrow\uparrow$
	$T_n \rightarrow$	$T_n \rightarrow$	$T_n \rightarrow$	$T_n \rightarrow$	$T_n \rightarrow$
ver 3	$k_r \downarrow\downarrow$	$k_r \downarrow$	$k_r \uparrow$	$k_r \uparrow$	$k_r \uparrow\uparrow$
	$T_n \rightarrow$	$T_n \rightarrow$	$T_n \rightarrow$	$T_n \downarrow$	$T_n \downarrow\downarrow$

TABLE I
DEVELOPMENT OF STRATEGY I

Strategy I

Since the rule base of [5] yielded partly acceptable results during previous studies, it is meaningful to retain the main concepts of the proposed tuning. Put in highly simplified terms, the fundamental idea is to diminish the damping (i.e. to increase the OS) by increasing the proportional gain k_r or by decreasing the integral time constant T_n. On the other side, the closed-loop damping can be increased by decreasing k_r or by increasing T_n. Bringing these basic principles in a table structure leads to the rule base illustrated in row "ver 1" of Table I.

In order to enable a free progression in the parameters space, it is meaningful to decouple the tuning for k_r and T_n. Hence, strategy I will focus on the proportional gain while

$d =$	BNEG	NEG	OK	POS	BPOS
ver 1	$k_r \rightarrow$	$k_r \rightarrow$	$k_r \rightarrow$	$k_r \rightarrow$	$k_r \rightarrow$
	$T_n \upuparrows$	$T_n \uparrow$	$T_n \rightarrow$	$T_n \downarrow$	$T_n \downdownarrows$
ver 2	$k_r \uparrow$	$k_r \uparrow$	$k_r \rightarrow$	$k_r \rightarrow$	$k_r \rightarrow$
	$T_n \upuparrows$	$T_n \uparrow$	$T_n \rightarrow$	$T_n \downarrow$	$T_n \downdownarrows$
ver 3	$k_r \uparrow$	$k_r \uparrow$	$k_r \downarrow$	$k_r \downarrow$	$k_r \downarrow$
	$T_n \upuparrows$	$T_n \uparrow$	$T_n \rightarrow$	$T_n \downarrow$	$T_n \downdownarrows$

TABLE II

DEVELOPMENT OF STRATEGY II

strategy II focuses on the integral time constant. In terms of strategy I, this leads to the configuration shown in the second row of Table I.

As presented in the requirements, there should always be a significant tuning action for at least one of the controller parameters. However, for the case d = OK there is no parameter modification wherefore the algorithm could easily be stuck. In order to solve this issue the proportional gain could be increased or decreased for the case d = OK. Since it is assumed that strategy I is very important for the tuning of overdamped cases, i.e. controllers with too small k_r, the proportional gain will be increased in this case.

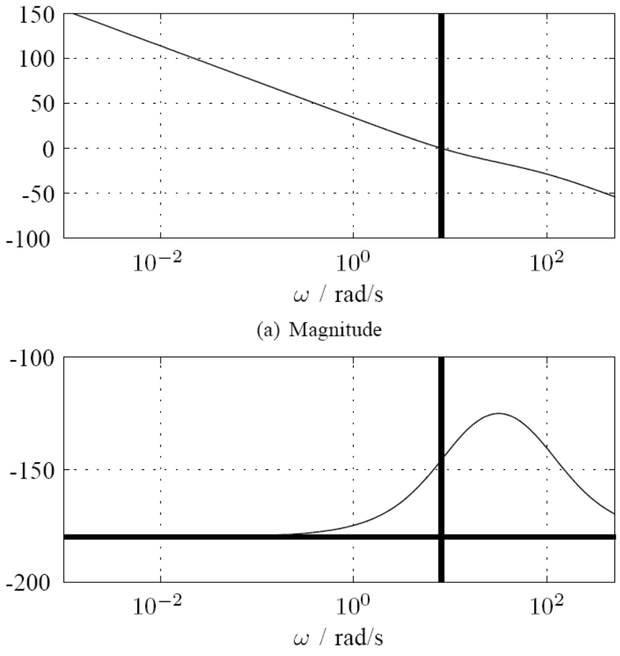
For overdamped systems, decreasing T_n will make the response faster and finally lead to the desired OS. Hence, these actions are also added to the rules for d = POS and d = BPOS in strategy I. The third row of Table I illustrates the final version of strategy I.

Strategy II

As strategy I is mainly directed to variations of k_r, the starting idea of strategy II is to focus on variations of T_n. When neglecting modifications of k_r, the initial rules in row "ver 1" of Table I give the rule base shown in row "ver 1" of Table II.

Again some modifications have to be included to comply with the requirements listed at the beginning of Section III. Several simulations and testbench results revealed that a combination of a too low proportional gain and a small integral time constant can lead to heavy oscillations in the step response. In those cases an increase of k_r can decrease the oscillations which is in contrast to strategy I and has therefore to be included in strategy II.

Fig. 4 exemplarily illustrates this behavior by means of the open-loop frequency response of a one-mass system being controlled by a PI controller. If k_r is increased, the magnitude plot moves upwards and the crossover frequency (bold vertical line) increases. This, in turn, will increase the phase margin as long as the crossover frequency approaches the maximum of the phase response. As a consequence, it would be meaningful to add a rule which increases the proportional gain in case of oscillating step responses as shown in the second row of Table II.



(a) Magnitude

(b) Phase

Fig. 4. Bode plot illustrating the rule d=NEG in strategy II

Particular processes suffer from particular restrictions, which has to be incorporated by an effective autotuning algorithm. Regarding drive systems, we illustrate how upcoming restrictions can be included to the rule base.

One of the most difficult controller autotuning situations in electrical drive systems is when the motor is in saturation for most of the time during a desired acceleration. This is mainly caused by a too high proportional gain and leads to a moderate damping and a turbulent steady state control. The only way to reduce the over-tightness of the control is to significantly decrease the proportional gain. Conventional autotuning algorithms fail to react adequately since existing system response characterizations are not capable of describing this situation. In consequence, the system behavior is inaccurately ascribed to another adverse controller setting, leading to implausible tuning action. In contrast, the algorithm proposed in this paper can be simply modified by decreasing k_r for damping values d = OK to d = BPOS in strategy II, which leads to the last row of Table II.

C. Step Response Characterization

The damping ratio d is the only input to the rule base in the proposed method. This section describes two approaches of determining d based on a step response.

The first way is to assume the step response of the closed-loop to be originated by a second order system (4). In discrete form, the system can be written as

$$G_z(z) = \frac{a}{1 + bz^{-1} + cz^{-2}}. \tag{7}$$

The parameters a, b and c can be estimated from a recorded step response, e.g. by applying the Nelder-Mead method. Assuming the discrete time system (7) to be originated from the continuous system (4) by using implicit Euler integration, the damping factor d can be calculated from the estimated parameters by

$$d = -\left(c + \frac{b}{2}\right)\sqrt{\frac{1}{c(1+b+c)}}. \qquad (8)$$

Regarding the closed-loop behavior, the method is limited to processes which can be described by second order systems. The procedure is robust to noisy measurement data but sensitive to unmodeled effects such as friction.

The second approach focuses on the maximum relative OS of the system response. If the reference step size is not exceeded by the system response, the system is regarded as overdamped and the damping factor is set to $d = 1$. Likewise, the damping is set to $d = 0$ if the system response exceedes twice the reference step size. In total, the damping factor can then be calculated by

$$d = \begin{cases} 1 & \text{, if } y_{max} < w_{ref} \\ 0 & \text{, if } y_{max} > 2 \cdot w_{ref} \\ \sqrt{\frac{(\ln \mathrm{OS})^2}{(\ln \mathrm{OS})^2 + \pi^2}} & \text{, else,} \end{cases} \qquad (9)$$

where the calculation for the last case follows directly from rearranging (5). In theory, this method is more sensitive to measurement noise, as it relies on a single sample of the step response. However, in simulations and testbench trials we obtained reliable characterization results if the noise level was not too high.

IV. Evaluation

Many simulations and testbench trials were conducted in order to investigate the performance of the proposed method w.r.t. motion control applications. A selection of the results are presented in this section. During all test trials the weighting factor was set to $r = 0.2$.

In the first example, the testbench is mechanically configured to resemble a one-mass system. A passive initial controller is applied. In each iteration of the method, a reference step from 0 to 200 rpm and a subsequent disturbance step are conducted. The results are shown in Fig. 5. The three subfigures show the nine iterations necessary to tune the controller. The first iteration illustrates the very slow dynamics regarding the reference and disturbance response. As there is no OS, the system is initially considered to be overdamped with $d = 1$. According to strategy I, k_r is increased and T_n is decreased. Step by step, these modifications improve the control performance during the next iterations. However, the first iteration in the second subfigure shows how the increased stiffness leads to inordinate controller action which becomes evident by the turbulent character of the stationary speed signal and an increased value of the performance index. Hence, the third tuning step is considered to be unsuccessful and the strategy is switched. After three iterations with strategy II and improving control quality, the second to last iteration has again a decreased performance index. The strategy is switched back to strategy I. However, the last iteration gives no improvement, wherefore the algorithm terminates by taking iteration 7 as the best tuning result.

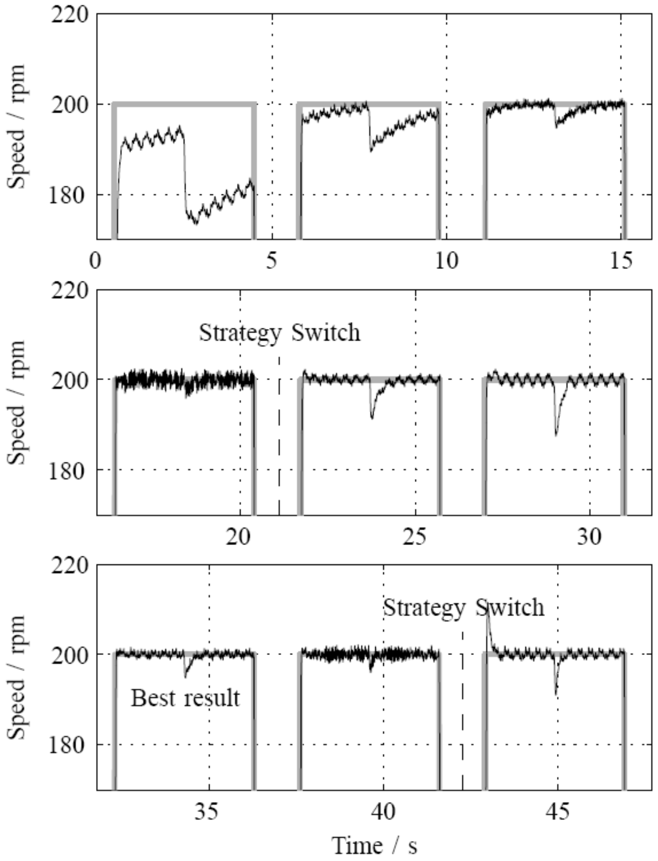


Fig. 5. Tuning protocol for slow initial controller

This type of test trial was repeated for a number of different initial controller parametrizations. The results are shown in Table III where $k_{r,0}$ and $T_{n,0}$ are the initial control parameters, k_r^* and T_n^* are the final controller parameters, *Iter* is the number of iterations and J is the final value of the performance index. Although the initial controller parameters are varied by a factor of 1000, the parameters converge to very similar values giving a good control performance. The only exception is case 6, where no improvement can be attained. This is due to the initial values which represent the stiffest controller realizable on the drive, i.e. the controller with the highest possible k_r and the lowest possible T_n. The corresponding tuning run is visualized in Fig. 6. The controller will create a step response with a damping in the range between “OK” and “BPOS”. However, for these values the rule base contains no strategy where k_r is decreased without simultaneously decreasing T_n. Hence, no improvement can be attained and the algorithm terminates after two iterations. The problem could be solved by modifying the rule base, e.g. by adding an additional strategy. However, note that the initial values of case 6 are not reasonable as they represent the stiffest possible configuration of the drive.

#	$k_{r,0}$	$T_{n,0}$	k_r^*	T_n^*	*Iter*	*J*
1	20	5	71.8	0.150	12	3.62
2	10	0.3	76.3	0.140	7	3.53
3	2	0.01	85.3	0.056	23	3.48
4	200	3	65.5	0.206	18	3.62
5	3.2	0.55	63.7	0.207	12	3.71
6	200	0.005	200	0.005	3	4.02
7	2	0.005	80.5	0.038	23	3.46
8	50	0.1	78.1	0.078	5	3.52
9	15	0.3	89.4	0.177	8	3.69
10	140	1	72.0	0.206	7	3.65

TABLE III

OVERVIEW OF TUNING RESULTS FOR DIFFERENT INITIAL CONTROLLERS

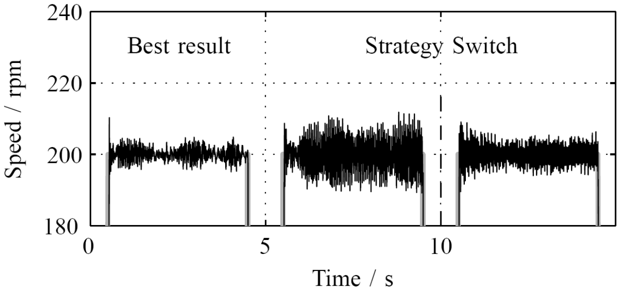


Fig. 6. Tuning protocol for testrun #6

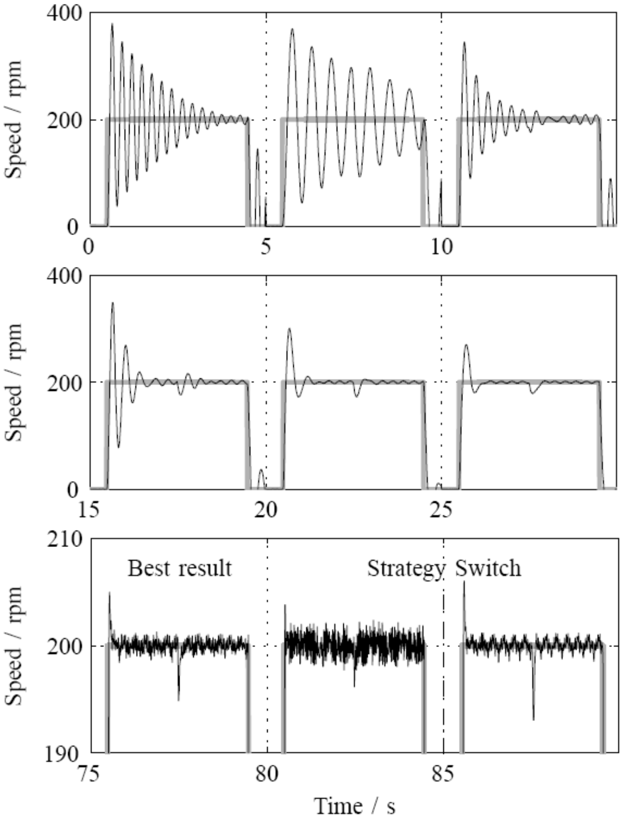


Fig. 7. Tuning protocol for testrun #7

An example for an initial controller with an oscillating step response is shown in Fig. 7. As the whole sequence takes relatively long to terminate (23 iterations), only the first six and the last three iterations are shown. Regarding the first two subfigures, it can be seen how the damping steadily increases. Note, that in the last subfigure the y-axis is scaled in order to visualize the differences between the last three iterations. Similar to the result in Fig. 5, the algorithm terminates when both strategies yield no improvement.

More testbench trials were conducted e.g. attaching a second mass to a load via a belt which made the system a two-mass system, using different reference and disturbance step sizes, all giving very promising results. Even simulation tests using models from process control like those found in [5] indicate that the method and rule base can be used for a wider class of systems than the initially intended one.

V. CONCLUSIONS

This paper addresses the design of a rule-based autotuning algorithm with focus on the application to drive systems. Compared to existing concepts, the approach of alternative tuning strategies enables the utilization of a compact and complete rule base. Due to the flexibility of the method, the ideas can be easily transferred to processes from other areas of application. The algorithm is able to deal with practical obstacles as measurement noise and nonlinearities occuring from process-specific non-idealities. Numerous trials in simulations and at a drive system testbench yielded adequate tuning results.

REFERENCES

[1] K. Inoue, J. Yoshitugu, S. Shirogane, P. Boyagoda, and M. Nakaoka. AC Servo Motor Drive Systems using Auto-Tuning Gain Parameter Processor with Automatic Learning Control Scheme. In *Proceedings of the IEEE International Symposium on Industrial Electronics*, 1997.

[2] I. Miki, T. Kumano, and T. Yamada. Auto-Tuning Method Based on Fuzzy Reasoning for Speed Controller in Vector-Controlled Induction Motor Drives. In *Conference Record of the 1993 IEEE Industry Applications Society Annual Meeting*, 1993.

[3] L. Mokrani and R. Abdessemed. A Fuzzy Self-tuning PI Controller for Speed Control of Induction Motor Drive. In *Proceedings of 2003 IEEE Conference on Control Applications*, 2003.

[4] B.-M. Pfeiffer and R. Isermann. Selftuning of Classical Controllers with Fuzzy-logic. *Mathematics and Computers in Simulation*, 37(2-3):101 – 110, 1994.

[5] H. Schaedel and R. Bartz. Fuzzy-Adaption von PI-Reglern im geschlossenen Regelkreis ohne Prozesskenntnis (German). In *9th Workshop "Fuzzy Control" of GMA-FA 5.22*, 1999.

[6] O. Wahyunggoro and N. Saad. Development of Fuzzy-logic-based Self Tuning PI Controller for Servomotor. *Advanced Strategies for Robot Manipulators*, 2010.

[7] J. Yoshitsugu, K. Inoue, and M. Nakaoka. Fuzzy Autotuning Scheme Based on α-parameter Ultimate Sensitivity Method for AC Speed Servo System. *IEEE Transactions on Industry Applications*, 36(2):492–499, 2000.

[8] Z.-Y. Zhao, M. Tomizuka, and S. Isaka. Fuzzy Gain Scheduling of PID Controllers. *IEEE Transactions on Systems, Man and Cybernetics*, 1993.