A Comparison of Tuning Methods for PID-Controllers with Fuzzy and Neural Network Controllers

Clemens Gross(✉) and Hendrik Voelker

Peter the Great St. Petersburg Polytechnic University, Saint Petersburg, Russia
{gross.yak, voelker.h}@edu.spbstu.ru

Abstract. Conventional approaches to control systems still present a reasonable solution for a variety of different tasks in control engineering problems. Controllers based on the PID approach are used in a wide range of applications due to their easy handling, realization and set up, as well as their modest need of computational resources during the runtime. In order to heuristically find near-optimal parameters for the controller design, different approaches to tuning PID controllers have been developed. The Ziegler–Nichols methods are still commonly used despite that they have long been known, though modern methods, such as the T-Sum method, have also emerged. In this work, a comparison of the tuned PID controllers with a Mamdani-Fuzzy-Logic controller and an adaptive neural network controller is offered. A unified step response is used to classify the performance of controllers. It is shown that a PID control can work just as well as a fuzzy logic or neural network control for simple applications with time-invariant parameters or in applications where the parameters only change slightly and no strict constancy of the plant output is necessary.

Keywords: Control engineering · PID · Fuzzy control · Neural network control · PID tuning · Fuzzy logic

1 Introduction

Although well known since the 1930s, the proportional–integral–derivative (PID) controllers are popularly used in many applications of control engineering even today. Huge efforts are made in order to optimize its control behaviour and to optimally adjust a PID controller's parameters to a specific use case. Therefore, classical PID control is still subject to many ongoing projects of applied research.

To optimally adjust the PID controller's parameters, an accurate model of the plant which should be controlled should be known or developed. Depending on the level of abstractness of a specific process, this step can pose a problem. Thus, different tuning methods for PID controllers exist. With the help of these methods, such as the Ziegler-Nichols (ZN) [1] or T-Sum [2] methods, it is possible to heuristically find close enough parameters to achieve a good performance of the process. Kumar et al. [3] as well as Wang et al. [4] compare in their works the various tuning paradigms with respect to

D. G. Arseniev et al. (Eds.): CPS&C 2019, LNNS 95, pp. 74–86, 2020.
https://doi.org/10.1007/978-3-030-34983-7_8

different plant and controller topographies. The current research offers an online adaption of PID controllers in dependence from time-variant parameters to overcome one of the major downsides in the PID control. Approaches to this have been available since the 1990s, as is shown, for example, by Sung et al. in [5]. Despite that, two modern and more sophisticated approaches for controllers are now used more and more often, to control not only nonlinear, time-variant processes. The Fuzzy Set Theory [6], or Fuzzy-Logic (FL) was first applied to control problems in 1974 by Mamdani [7], and a general overview of the possible industrial applications of Fuzzy Control as compared to PID controllers is given in [8]. By its help the system behaviour and the desired control reaction can be described by linguistic variables and, therefore, is easy to adapt to a plant the exact parameters of which are unknown. The second modern control approach is based on Artificial Neural Networks (ANN). Even though the theory behind simple ANN is also fairly old, first postulated in 1943 by McCulloch and Pitts [9], only recently it was actually applied to different problems in engineering and information technology. Different principles of Neural Network Control (NNC) have evolved since the 1990s as is shown in [10]. Current research in control theory focuses on merging the advantages of classical PID control with FL (Deng et al. [11]) and ANN approaches in order to adapt PID parameters during the process runtime. Application of NNC to robotic manipulators is shown in [12] by Shuzhi et al., whereas Potekhin et al. [13] investigate a Fuzzy Neural Network for controlling autonomous decentralized energy grids.

This work shows a comparison between the PID tuning methods, FL and ANN control. The aim of this work is to indicate which of the methods has the best performance as along with the lowest implementation effort. For the PID tuning methods, the ZN and T-Sum ones, are chosen, while a Mamdani-FL controller and another one based on Feed-Forward-ANN is used for simulation of the controllers and the process.

2 Different Control Approaches

Three classes of controllers are compared in this work. PID Controllers and those based on FL and ANN are the main control paradigms used in closed-loop control applications. This section gives an overview of the theory and systematic schemes behind the different types of controllers and their respective tuning methods.

2.1 PID-Control

A PID controller is composed of three individual parts, hence is the name of proportional–integral–derivative controller. Linking these three blocks in parallel results in the general transfer function for a PID controller in the Laplacian domain

$$K_{PID}(s) = K_P + \frac{K_I}{s} + K_D s = K_P\left(1 + \frac{1}{T_I s} + T_D s\right), \tag{2.1}$$

where $T_I = K_P/K_I$ is called the reset time and $T_D = K_D/K_P$ is called the lead time [14]. By transforming Eq. (2.1) back to the time domain as shown in Eq. (2.2), the dependence of the controller's output $u(t)$ from the control error $e(t)$ is easily derived:

$$u(t) = K_P e(t) + \frac{K_P}{T_I}\int_0^t e(\tau)d\tau + K_P T_D \frac{de(t)}{dt}. \tag{2.2}$$

A graphical presentation of a PID controller with the control error and control output in the Laplacian domain can be seen in Fig. 1. If the plant is well known and can be modelled mathematically without further difficulties, numerical methods to set the parameters are preferred. However, it is not always practical to invest a lot of time and resources into developing an adequate mathematical model of the respective plant. Therefore, different tuning methods exist which make it possible to heuristically find near-optimal solutions for the parameters of the PID controller.

2.1.1 Ziegler-Nichols Tuning Method

While the Ziegler-Nichols method [1] has been known since 1942, it is still used today, mainly for strongly delayed processes. The ZN method proposes two different approaches; one of the two being presented here. The plant is approximated as a first-order plus time delay model. The time delay of the plant T_{dead}, the time constant T and the stationary amplification K_s have to be known or determined experimentally from the system's step response by adding an inflexion tangent to the step response of the system. The time between the zero point of the time scale, when the step was applied to the plant, and the intersection of the inflexion tangent with the time axis can be considered equal to T_{dead}, while the time between the intersection of the inflection tangent with the x-axis and with the stable output of the plant is equal to T. The PID controllers' parameters are then adjusted in the following way: $K_P = 1.2/K_s \cdot T/T_{dead}$, $T_I = 2 \cdot T_{dead}$ and $T_D = 0.5 \cdot T_{dead}$. Other tuning methods, like Chien-Hrones-Reswick [15], are based on the second method of ZN.

2.1.2 T-Sum Tuning Method

The T-Sum method was first introduced in 1995 by Kuhn [2]. Together with other approaches, like the one proposed by Åström and Hägglund [16], it stands for a modern and more sophisticated approach to PID tuning. It can be used for plants which can be characterized by a low-pass behaviour and have the transfer function as:

$$G(s) = K_s \frac{(1 + T_{U,1}s)(1 + T_{U,2}s)\ldots(1 + T_{U,m}s)}{(1 + T_{L,1}s)(1 + T_{L,2}s)\ldots(1 + T_{L,n}s)} e^{-sT_{dead}} \tag{2.3}$$

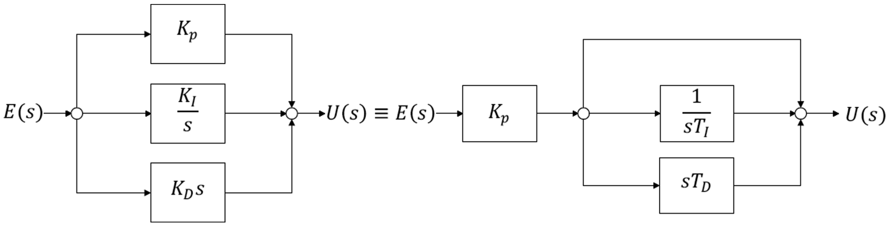

Fig. 1. Block diagram of a PID controller

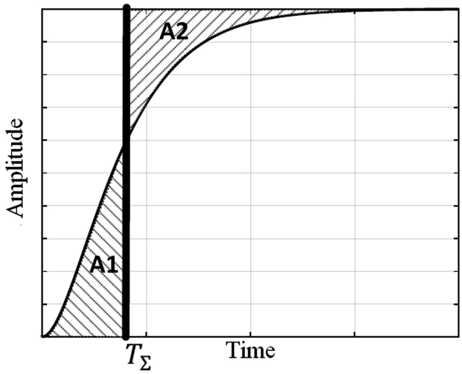

Fig. 2. Graphical determination of the T-Sum

Table 1. Parameters for the T-Sum method (Original)

Parameter	Value
K_P	K_s
T_l	$0.66 \cdot T_\Sigma$
T_D	$0.167 \cdot T_\Sigma$

Table 2. Parameters for the T-Sum method (Fast)

Parameter	Value
K_P	$2/K_s$
T_l	$0.8 \cdot T_\Sigma$
T_D	$0.194 \cdot T_\Sigma$

The T-Sum tuning method is named after the main operation necessary to adjust the PID parameters with its help, the sum of the delaying time constants T_L in the numerator of Eq. (2.3) minus the deriving time constants in the denominator and plus the dead time:

$$T_\Sigma = T_{dead} + \sum_{i=1}^{n} T_{L,i} - \sum_{j=1}^{m} T_{U,j}. \tag{2.4}$$

If the transfer function of the plant is unknown, the T-Sum shown in Eq. (2.4) can also be obtained from the experimentally found step response of the plant, either arithmetically (2.5)

$$T_\Sigma = \int_0^\infty \left(1 - \frac{y(t)}{K_s}\right) dt \tag{2.5}$$

or graphically from the step response of the system, which is denoted as $y(t)$ in (2.5). Graphically, the value of T_Σ can be derived by introducing a straight line, perpendicular to the time axis. This straight line divides the area between it and the y-axis under the step response and above the x-axis from the area above the step response and below the steady state of it between this straight line and the steady state of the step response. When these two areas are equal ($A_1 = A_2$ in Fig. 2), the value for T_Σ can be directly seen from the position of this straight line on the time axis, as shown in Fig. 2. The values for the PID's parameters can then be directly seen from Table 1 the original T-Sum method and from Table 2 the fast T-Sum method, which compromises a higher overshoot for a faster settling time.

2.2 Fuzzy Control

Unlike PID controllers, which are parametrized either by mathematical modelling or experimental knowledge, fuzzy control relies more on previous knowledge in terms of heuristic *IF <condition> THEN <action>* rules. Therefore, designing a fuzzy controller is actually similar to the experimental method for constructing a table of inputs and corresponding output values called a ruleset in Fuzzy Logic. What makes FL interesting, though, is that due to the fuzzification of the inputs, not only the situations actually listed in the respective ruleset result in an action, but also all intermediate values which at least partially correspond to one of the if-conditions. To achieve this, the reference input $e(t)$ of the FL controller is mapped to a value in the range [0, 1].

This certain value is called degree of belief in FL and is achieved by applying the fuzzy membership functions (see Fig. 3) to the crisp input $e(t)$. A typical FL controller is composed of the fuzzification of the reference input, which is then led to the Fuzzy Inference System (FIS). Depending on the respective if-then rules in the rule base, the FIS infers a certain fuzzy control input $u(t)$, which is then defuzzified and applied to the plant of the control loop [17]. This can be seen schematically in Fig. 4.

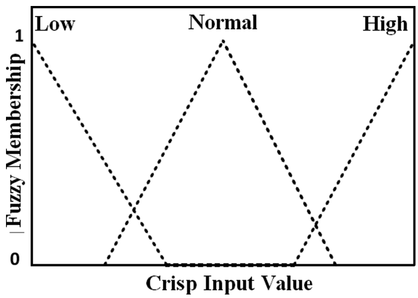

Fig. 3. Exemplary membership functions

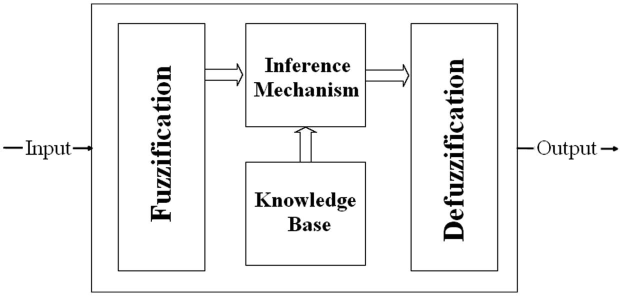

Fig. 4. Schema of a fuzzy controller

A big advantage of fuzzy controllers is their ability to work with plants with uncontinuous transfer functions and their robustness when a plant with time-variant behaviour is controlled [17]. The first applications of FL to control problems were proposed by Sugeno in [18].

2.3 Neural Network Control

Neural Network Control is another approach to control systems that has emerged lately from the huge interest in ANN. In the general ANN research, ANN are classified by their structure, which are feed-forward, radial-basis functions, convolutional or long-short term memory ANN among others. For NNC, though, Agarwal [10] tried to classify different controllers. The basic principle of ANN relies on small, independent neurons, inspired by the biological processes of the brain of living organisms, which multiply n input signals, by the respective weights $w_1, \ldots, w_n$. The weighted input signals are then summed up with a possible bias of the neuron and fed to a specific activation function, which gives back the output of the neuron. Literature sources, e.g., [19] or [20], present a more detailed introduction to ANN.

One of the main advantages of NNC is the ability of feed-forward ANN with nonlinear, differentiable activation functions to approximate any given function, even if the function is nonlinear [21]. This is used in the first step towards building a NNC, when the plant's transfer function is approximated with the help of ANN, instead of being analytically modelled manually. The second step, the actual NNC, profits from one of the main reasons why the ANN are so popular nowadays: i.e., their ability to learn. Due to this fact, it is possible to build adaptive controllers with the help of ANN [22]. However, the necessity of the structure of the ANN to be determined a-priori, the control task can pose a problem because it can lead to an overdetermined network structure resulting in high computational complexity, or to an underdetermined structure causing poor performance of an actual controller [21].

3 Experimental Set-Up

In the course of this work, three different control paradigms are simulated in Matlab/Simulink R2016B. The control reference $e(t)$ is implemented by a unified step at the time $t = 0$. The obtained unified step response of the plant for different controllers is then compared.

3.1 Plant

In order to compare the performance of the controllers, a plant for the control loop should be chosen. Due to its wide range of applications, a DC motor is selected; it can be modelled according to the Newton (3.1) and Kirchhoff (3.2) laws

$$J\frac{d^2\theta}{dt^2} = t - b\frac{d\theta}{dt}; \tag{3.1}$$

$$L_a\frac{di}{dt} = -R_a i + V - e \tag{3.2}$$

as a second order plant. A DC motor can be modelled in more detail as well, the representation in the form of a linear PT2 element is chosen due to the possibility to model a variety of different control problems in this form of a second order plant. Figure 5 represents the model of the plant which is created in Simulink. In order to study the impact that time-variant behaviour of the plant produces on the different controllers, the values of the plant parameters are altered after the controllers have been adjusted to it. This adjustment lies in the range of plus or minus 20% for each parameter and is, therefore, by all means within the range of deviations that the parameters of individual components of the same series may have or may acquire over the time [23]. The changes applied to the plant parameters after the optimization of the controllers can be seen in Table 3.

3.2 Controllers

A PID controller is realized by the use of respective gain blocks, integrators and derivative blocks, just as shown in Fig. 1 The different parameters of the PID controller are then derived from the step response of the plant, using the 2nd ZN method, as well as the original and fast adjustment values from the T-Sum method. This results in the following parameters for PID controllers that are shown in Table 4.

Table 3. Change of plant parameters to simulate change over the time

$L_{a,new} = 0.9 \cdot L_a$	$R_{a,new} = 0.9 \cdot R_a$	$K_{t,new} = 1.2 \cdot K_t$
$J_{new} = 0.8 \cdot J$	$b_{new} = 1.1 \cdot b$	$K_{e,new} = 1.05 \cdot K_e$

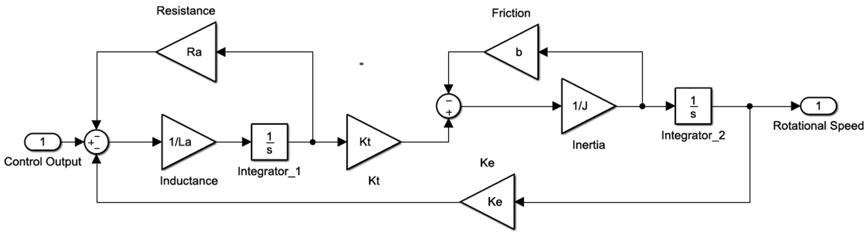

Fig. 5. DC motor modelled as a linear second order plant in simulink

The fuzzy controller as well as the neural network controller is chosen from the respective toolbox in Simulink. For the fuzzy controller, a Mamdani approach with two input variables (control reference $e(t)$ and fed-back plant output $y(t)$) and the control output $u(t)$ is applied. Five triangular membership functions (a positive and negative ones for the control reference and a positive, negative and zero ones for the feedback signal) and three output membership functions are implemented with a respective ruleset. The membership functions for the input and output variables can be seen in Fig. 6.

Similarly, a controller from the neural network toolbox in Simulink is chosen to characterize the implementation and the behaviour of such a type of controller. The chosen NN predictive controller [24] has a specific control horizon of discrete steps in which the plant's behaviour is predicted and the necessary control actions are performed (the control horizon). These parameters are set to ten and eight respectively. The controller is trained by random control reference inputs in a given range of values and time. During a training session with the presented plant, a series of 5000 discrete time steps is run over the NN with a batch size of 10. The layers consist of eight neurons each, and the weights are optimized by using a 1-dimensional backtracking algorithm in Matlab. Figure 7 shows the architecture of the NN used in the controller.

Table 4. Parameters for the PID controller

Parameter	2^{nd} Ziegler-Nichols	T-Sum (original)	T-Sum (fast)
K_p	4.505	1.000	2.000
T_I	1.334	1.815	2.200
T_D	0.333	0.459	0.535

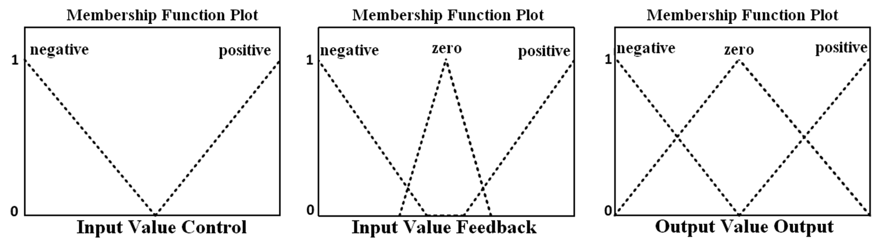

Fig. 6. Membership functions of the implemented fuzzy controller

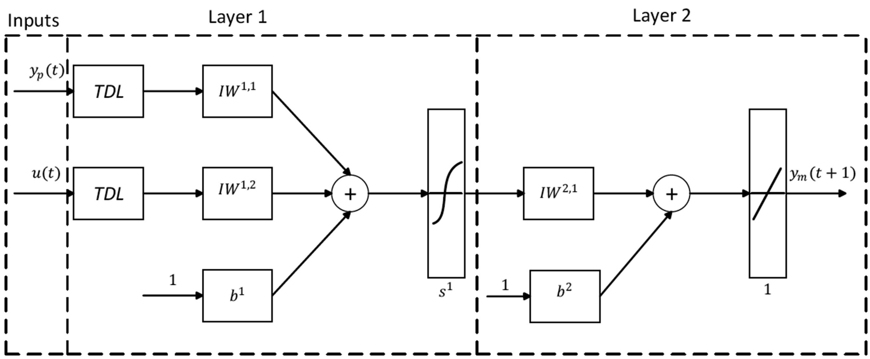

Fig. 7. Architecture of the NN predictive controller

4 Results

The simulation is first carried out with the plant's parameters for which the controllers were optimised and it can be seen in Fig. 8. In order to assess the performance of different controllers, a unified method to quantify the plant's output is necessary. Important parameters used to evaluate controller's performance with the use of the step response are the overshoot of the plant's output $y(t)$ over the reference signal $e(t)$, the settling time (in dependence from a certain error criteria), and the stationary error. In this work, the settling time is evaluated with respect to a deviation of 0.8% from the stationary signal. In order to include all of these parameters in the estimation, the root mean squared error (RMS) of the discrete signals y and e (4.1) is used:

$$\mathrm{RMS} = \sqrt{\frac{\sum_{i=1}^{n}(e_i - y_i)^2}{n}}. \tag{4.1}$$

The RMS, the overshoot and the settling time are shown in the following Table 5 for different controllers. None of the controllers showed a stationary error. After the plant's parameters were altered, the simulation is carried out again. The results can be seen in Fig. 9 and Table 6 respectively. In order to be able to quantify the influence that the change of the plant's parameters has on the controller's performance, the difference of the RMS from Table 5 RMS_{old} and the RMS with the changed plant's parameters RMS_{new} is computed (4.2):

$$\Delta_{\mathrm{RMS}} = \frac{\mathrm{RMS}_{old} - \mathrm{RMS}_{new}}{\mathrm{RMS}_{old}} \cdot 100\% \tag{4.2}$$

It can be seen that the results of the different control paradigms differ. In view of the minimal error criteria, the ZN-tuned PID controller shows the best result, even though it has the highest overshoot exceeding those of the controller with the parameters set by the fast T-Sum method by more than the factor three. The FL controller shows the best behaviour in terms of the overshoot but in terms of the RMS error criteria it is at the bottom together with the NN predictive controller. This would not even change when the plant's parameters are changed to simulate time-variant plant behaviour. If a low overshoot is required, though, the examined FL controller has a feasible behaviour.

Table 5. Key performance indicators for the controllers, original plant parameter

Parameter	PID Control			Fuzzy Control	NN pred. Control
	2nd ZN	T-Sum (or.)	T-Sum (fast)		
RMS	0.232	0.301	0.259	0.312	0.317
Overshoot [%]	40.65	6.81	12.37	2.39	6.65
Settling t. [s]	8.12	11.4	8.8	10.24	9.24

Table 6. Key performance indicators for the controllers, changed plant parameter

Parameter	PID Controller			Fuzzy Controller	NN pred. Control
	2nd ZN	T-Sum (or.)	T-Sum (fast)		
RMS	0.212	0.285	0.244	0.285	0.293
Δ_{RMS} [%]	8.352	5.357	5.946	8.665	7.556
Overshoot [%]	35.52	0.46	7.43	6.79	11.47
Settling t. [s]	7.05	10.4	7.95	8.41	10.41

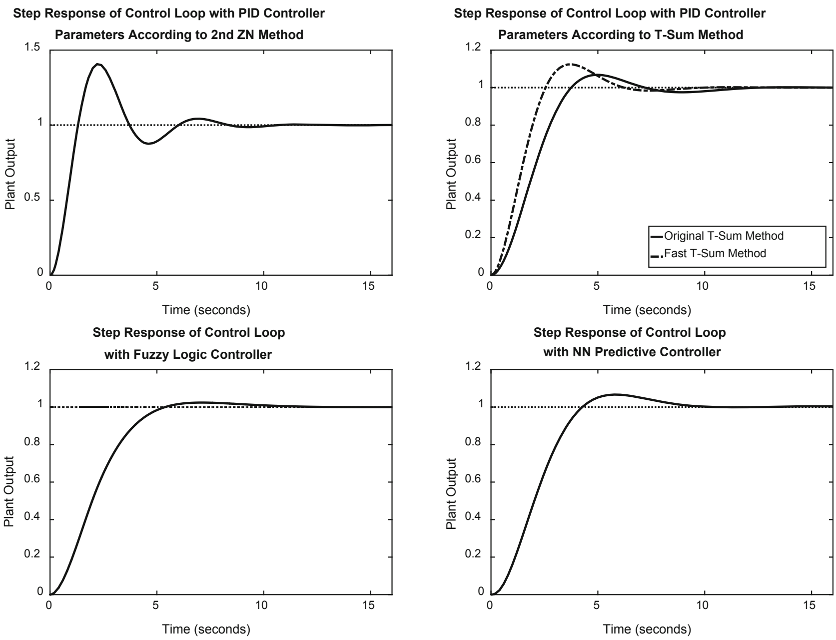

Fig. 8. Comparison of the step responses of the control loop with the original parameters

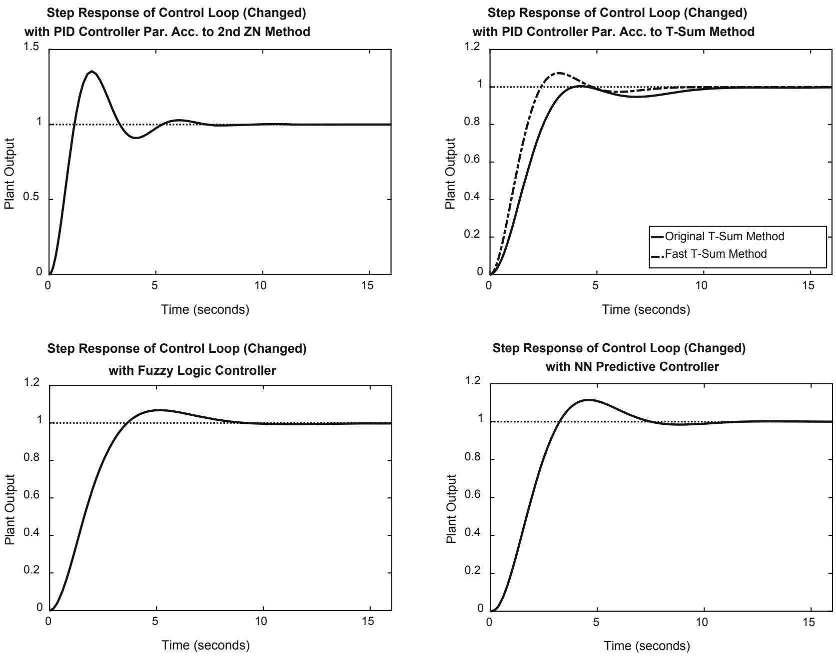

Fig. 9. Comparison of the step responses of the control loop with the changed parameters

5 Conclusion

Controllers – and different tuning methods – can be compared to each other by the basic criteria that are mandatory for every control system. All of the tested controllers showed stable behaviour, had good accuracy as measured by the resting control deviation, and could be considered robust, as all the controllers were able to satisfactorily control the plant, even when the parameters of the plant altered over time. The coherence of speed and attenuation of control loops is shown in this work again. The controllers which have a lower setting time, as a general rule, have a higher overshoot. Therefore, a trade-off has to be made when choosing the right controller for a control application. For the set-up used in this work, the ZN-tuned PID controller showed the fastest settling time, whereas the FL controller had the lowest overshoot.

It has to be said, though, that the behaviour of the FL controller highly depends on the chosen ruleset and membership functions, and a different characteristic for the step response can, therefore, be obtained with an FL controller. This is also true for the NN predictive controllers, where the chosen architecture of a controller, as well as the training data, influences the future control action. Therefore, these two certain types of controllers stand only prototypical for the respective control approaches, as no generalized methods of bestpractice design of these controllers exists so far.

It should be noted that for small control problems that can be linearized with sufficient accuracy, like the one examined in this work, neither a fuzzy controller nor a neural-network controller are of certain advantageous character. In all performance indices shown in Tables 5 and 6, the PID controller with a specific tuning method can keep up with more sophisticated control approaches. By choosing other tuning methods for a PID controller and manual fine tuning of the parameters, the shown behaviour can be changed if necessary. Especially when taking into account the lesser implementation expenses and the lower computational complexity of a PID controller, fuzzy and NN controllers cannot be considered practicable for such a control application.

References

1. Ziegler, J.G., Nichols, N.B.: Optimum settings for automatic controllers. Trans. ASME **64**, 759–768 (1942)
2. Kuhn, U.: Eine praxisnahe Einstellregel für PID-Regler: Die T-Summen-Regel. Automatisierungstechnische Praxis **5**, 10–16 (1995)
3. Kumar, R., Singla, S., et al.: Comparison among some well known control schemes with different tuning methods. J. Appl. Res. Technol. **13**, 409–415 (2015)
4. Wang, L., Freeman, C., Roger, E.: Experimental evaluation of automatic tuning of PID controllers for an electro-mechanical system. IFAC PapersOnLine **50**(1), 3063–3068 (2017)
5. Sung, S.W., Lee, I.-B.: On-line process identification and PID controller autotuning. Korean J. Chem. Eng. **16**(1), 45–55 (1999)
6. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**(3), 338–353 (1965)
7. Mamdani, E.H.: Application of fuzzy algorithms for control of simple dynamic plant. Proc. Inst. Electr. Eng. **121**(12), 1585–1588 (1974)
8. Lugli, A.B., et al.: Industrial application control with fuzzy systems. Int. J. Innov. Comput. Inf. Control **12**(2), 665–676 (2016)

9. McCulloch, W., Pitts, W.: A logical calculus of ideas immanent in nervous activity. Bull. Math. Biophys. **5**(4), 115–133 (1943)
10. Agarwal, M.: A systematic classification of neural-network-based control. IEEE Control Syst. Mag. **17**(2), 75–93 (1997)
11. Deng, J., et al.: Self-tuning PID-type fuzzy adaptive control for CRAC in datacenters. In: Springer IFIP Advances in Information and Communication Technology (AICT), vol. 419, no. 1, pp. 215–225 (2014)
12. Shuzhi, S.G., Hang, C., Woon, L.: Adaptive neural network control of robot manipulators in task space. IEEE Trans. Industr. Electron. **44**(6), 746–752 (1997)
13. Potekhin, V.V., Pantyukhov, D.N., Mikheev, D.V.: Intelligent control algorithms in power industry. EAI Endorsed Trans. Energy Web **3**(11), e5 (2017)
14. Lunze, J.: Regelungstechnik 1 – System-theoretische Grundlagen, Analyse und Entwurf einschleifiger Regelungen. Springer, Heidelberg (2010)
15. Chien, K.L., Hrones, J.A., Reswick, J.B.: On the automatic control of generalized passive systems. Trans. ASME **74**, 175–185 (1952)
16. Åström, K.J., Hägglund, T.: Automatic tuning of simple regulators with specifications on phase and amplitude margins. Automatica **20**(5), 645–651 (1984)
17. Passino, K.M., Yurkovich, S.: Fuzzy Control. Addison Wesley Longman Inc., Menlo Park (1998)
18. Sugeno, M.: Industrial Applications of Fuzzy Control. Elsevier Science Inc., New York (1985)
19. Haykin, S.: Neural Networks - A Comprehensive Foundation. Pearson Education (Singapore) Pte. Ltd., Delhi (2005)
20. Patterson, J., Gibson, A.: Deep Learning - A Practicioner's Approach. O'Reilly Media Inc, Sebastopol (2017)
21. Liu, G.: Nonlinear Identification and Control - A Neural Network Approach. Springer, London (2001)
22. Rovithakis, G.A., Christodoulou, M.A.: Adaptive Control with Recurrent High-Order Neural Networks. Springer, London (2000)
23. Halfmann, C., Holzmann, H.: Adaptive Modelle für die Kraftfahrzeugdynamik. Springer, Heidelberg (2003)
24. Soloway, D., Haley, P.: Neural generalized predictive control. In: Proceedings of the 1996 IEEE International Symposium on Intelligent Control, Dearborn, MI, USA (1996)