University of Ljubljana

J. Stefan Institute, Ljubljana, Slovenia

Report **DP-7286**

A comparison between different PI controller tuning methods

Damir Vrančić
Youbin Peng*
Christian Danz**

*Department of Control Engineering
Free University of Brussels, Brussels, Belgium

**Technical University Berlin

November 1995

Table of contents

1. Introduction

There are variety of controller tuning methods available. In general, tuning methods can be separated into 4 main groups [Haalman, 1966]:

- methods based on the Bode diagram
- methods based on the Nyquist diagram
- methods based on the process radiation curve (step response of the process)
- methods based on the process transfer function

Controller parameter adjustments based on the Bode diagram, the Nyquist diagram and the process transfer function, require information about the process model in frequency domain, while adjustments based on the process radiation curve need only process time responses on particular input excitation signals.

In this report we will present the results of the tuning of the PI controller for 9 different processes models at 7 different process constants. The results are performed by using 4 different tuning methods. One method is based on the Nyquist diagram (frequency response method) and the other three are based on the process radiation curve.

2. Background materials

The controlled system is shown in Fig. 1, where *w*, *e*, *d* and *y* denote process set-point, control error, process disturbance and process output, respectively. Controller parameters K and T_i denote proportional gain and integral time constant, respectively.

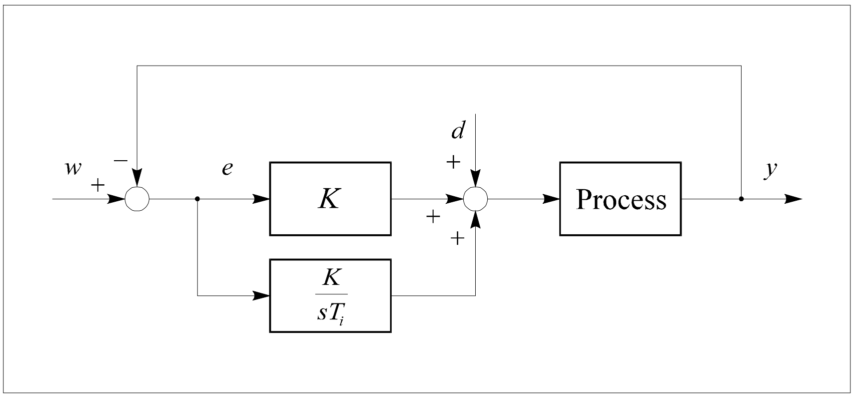

Fig. 1. The closed loop system with PI controller

Four different tuning methods were used to determine controller parameters:

- Ziegler-Nichols (ZN) adjustments based on the process radiation curve
- Cohen-Coon (CC) adjustments
- Chien-Hrones-Reswick (CHR) adjustments
- frequency response method (FRM) [Hanus, 1975]

The first three methods require the process step response (process radiation curve).

Typical process step response is shown in Fig. 2.

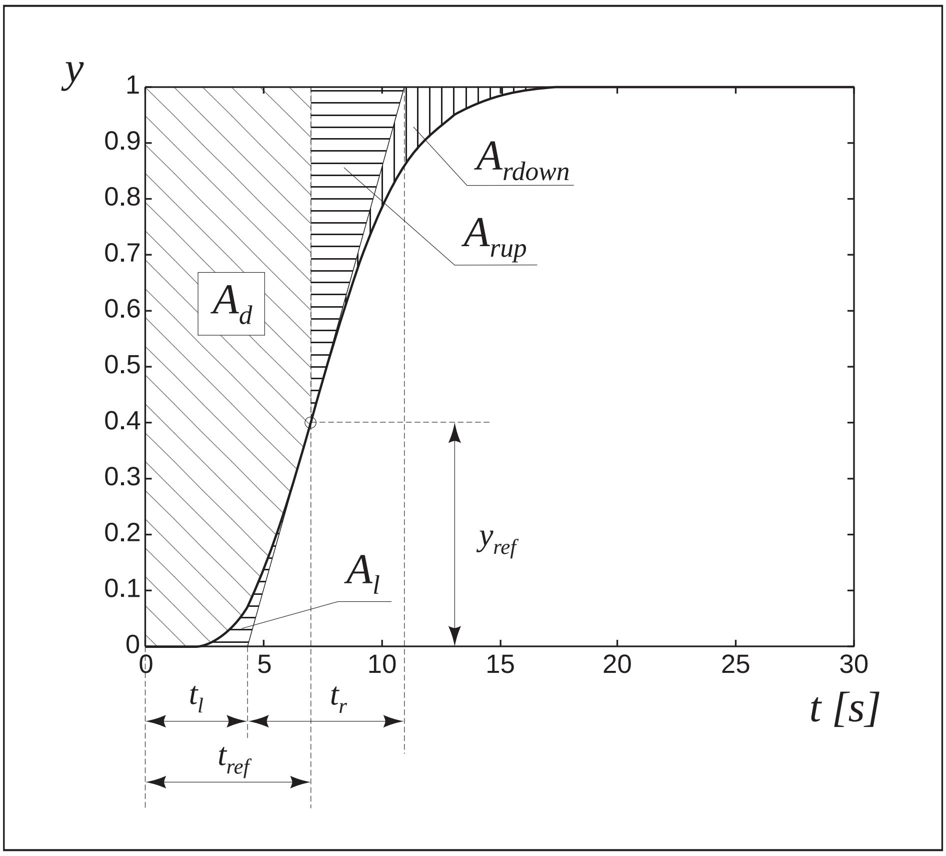

Fig. 2. Typical process step response

Variables t_l, t_r, t_{ref}, y_{ref} denote process lag time, rise time, reflection time and reflection amplitude, respectively. The circle denotes the reflection point. In Fig. 2 there are also areas marked with A_d, A_l, A_{rup} and A_{rdown}. They are added for further theoretical evaluations of tuning results. All indicated variables will be measured in each experiment.

Nevertheless, first three tuning methods need only the information about process lag time (t_l) and process rise time (t_r).

The last method (FRM) requires the whole process transfer function for controller tuning.

3. Tuning procedures

3.1. Ziegler-Nichols adjustments for PI controller

The ZN tuning rules are among the oldest known rules for tuning PI(D) types of controllers. Nevertheless, they are still widely used. In the last decades, several modifications of the original rules have been proposed by [Åström and Hägglund, 1995] [Hang et al., 1991] [Hang and Cao, 1993].

In our experiments, the original ZN tuning rules, based on the process time response, will be used [Haalman, 1966]:

$$K = 0.9\frac{t_r}{t_l}, \quad T_i = 3.3t_l \tag{1}$$

where values t_r and t_l are the process rise and process lag time, respectively (see Fig. 2).

3.2. Cohen-Coon adjustments for PI controller

The CC rules for tuning the PI controller are based on the following criteria: 25% damping ratio (see e.g. [Vrančić et al., 1993]) and minimum offset, whilst the adjustment of the rate action is a compromise between the period of damped oscillations and the minimum of the integral of *e* (control error) [Haalman, 1966].

The Cohen-Coon tuning rules for PI controller are the following:

$$K = \frac{1}{12} + \frac{9}{10}\frac{t_r}{t_l}, \quad T_i = t_l\frac{30t_r + 3t_l}{9t_r + 20t_l} \tag{2}$$

3.3. Chien-Hrones-Reswick adjustments for PI controller

There are different CHR rules available for tuning the PI controller. They depend on whether we like to have aperiodic response or 20% process overshoot, both for regulation or tracking purposes [Šega, 1991].

We used the specific CHR rules which give 20% process overshoot in the closed-loop tracking mode:

$$K = 0.6\frac{t_r}{t_l} \qquad T_i = t_r \tag{3}$$

3.4. Frequency response method for PI controller

The typical shape of the process Nyquist curve $G_p(j\omega)$ is shown in Fig. 3.

The main idea of the frequency response method (FRM) is to find such a controller $G_c(j\omega)$ which will drive the open-loop response of $G_p(j\omega)G_c(j\omega)$ toward the vertical line $Re\{G_p(j\omega)G_c(j\omega)\}=-1/2$, as shown in Fig. 4. The FRM method has another restriction: $Re\{G_p(j\omega)G_c(j\omega)\} \geq -1/2$ and $Re\{G_p(0)G_c(0)\} = -1/2$. The reason for such limitations lies in the fact that in such case the *closed-loop* response (see M and N circles [DiStefano et al., 1990]) will have the *unchanged gain* (M circle is 1) at low frequencies, and the gain will *decrease* at higher frequencies. Therefore, there will be no resonance peak in closed-loop amplitude response, so the system will be critically damped [DiStefano et al., 1990] [Boucher and Tanguy, 1976] and no oscillations will exist in closed-loop time response.

There is another advantage of the frequency response method. The given frequency limitations will also assure the stability of the system. The amplitude margin will always be greater than or equal to 2 ($A_m \geq 2$) and the phase margin greater than or equal to 60 degrees ($\varphi_m \geq 60°$) (see e.g. [Vrančić and Peng, 1994]).

Parameters can be adjusted in different ways. One of the possible concepts is using the optimisation and the second one is the analytical concept. While the latter is rather complex (see [Boucher and Tanguy, 1976]), we were finding the PI parameters by optimisation. At first T_i was given some small value (e.g. 0.1) and K=1. Then T_i was increased such that $Re\{G_p(0)G_c(0)\} = -\alpha$ and $Re\{G_p(j\omega)G_c(j\omega)\} \geq -\alpha$ for $\forall\ \omega$. The maximum T_i, for which the last conditions are still fulfilled, is the solution. K is then changed such that $Re\{G_p(0)G_c(0)\} = -1/2$:

$$K = \frac{0.5}{\alpha} \tag{4}$$

The optimisation procedure is made in program package MATLAB. It is called FINDKPTI.M and is given in the appendix.

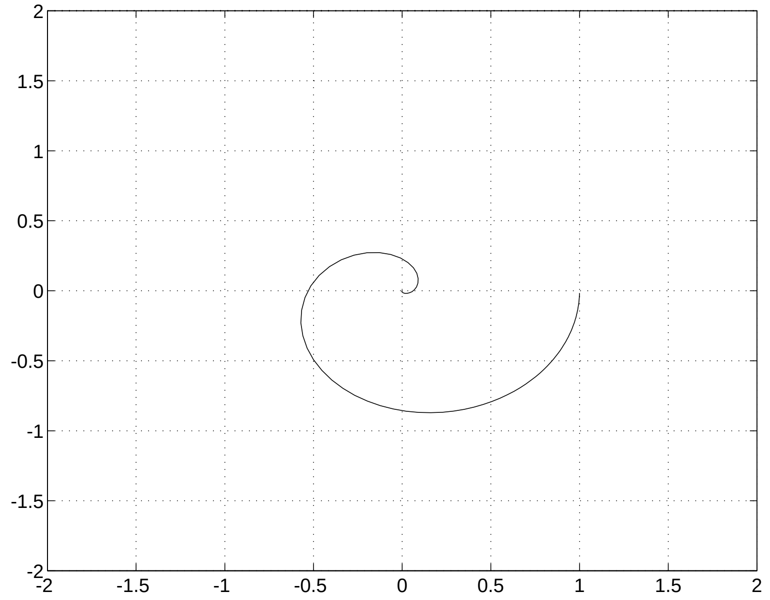

Fig. 3. Typical process Nyquist curve $G_p(j\omega)$

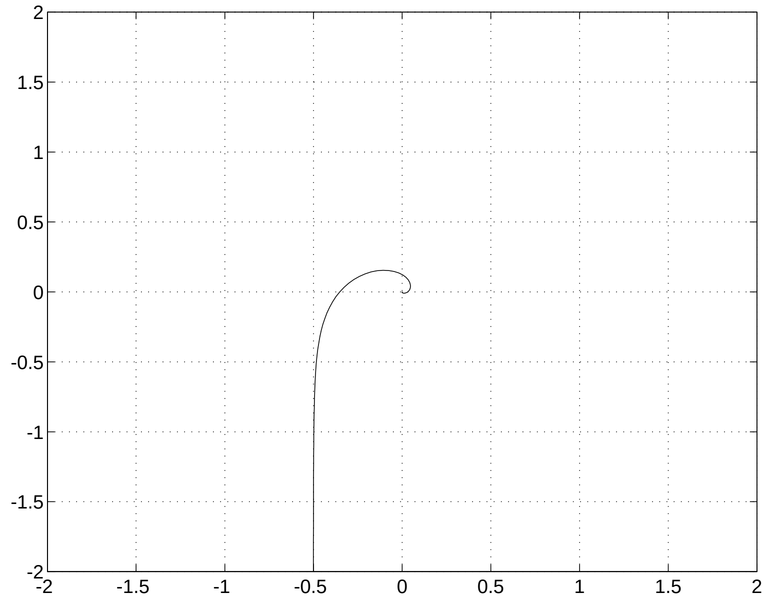

Fig. 4. The Nyquist curve of $G_p(j\omega)*G_c(j\omega)$

4. Open-loop experiments with parameter calculations

Here, we will show the specific process values (t_l, t_r, t_{ref}, y_{ref}, A_d, A_l, A_{rup} and A_{rdown}), which are found from the process step response, and the calculated PI parameters (K and T_i) by using four different tuning methods. The MATLAB procedure FINDALL.M was used for calculation and is given in appendix.

The process step response was obtained by running the SIMULINK scheme shown in Fig. 5.

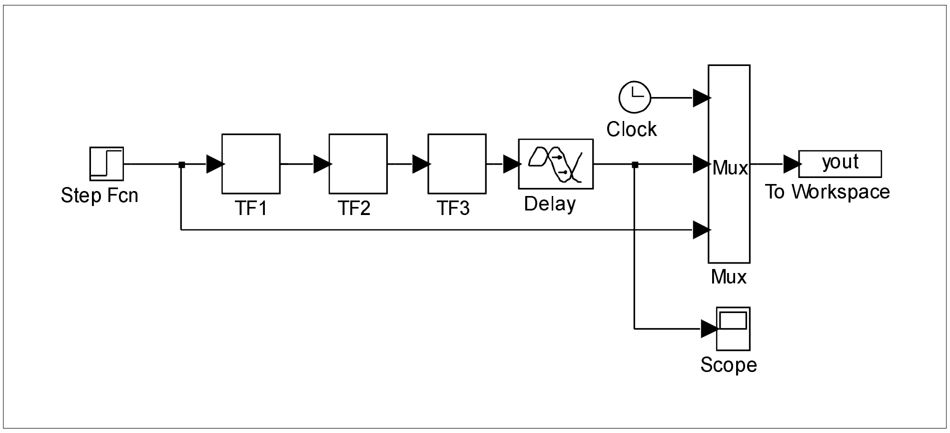

Fig. 5. The SIMULINK scheme for getting the process step response

Blocks TF1*TF2*TF3 represent the process transfer function.

Process 1:

$$G_{P1} = \frac{e^{-s}}{(1+sT)}$$

T	t_r	t_l	t_{ref}	y_{ref}	A_d	A_l	A_{rup}	A_{rdown}
0.1	0.1	1.0	1.0	0.0	1.0	0.0	0.05	0.05
0.2	0.2	1.0	1.0	0.0	1.0	0.0	0.1	0.1
0.5	0.5	1.0	1.0	0.0	1.0	0.0	0.25	0.25
1	1.0	1.0	1.0	0.0	1.0	0.0	0.5	0.5
2	2.0	1.0	1.0	0.0	1.0	0.0	1.0	1.0
5	5.0	1.0	1.0	0.0	1.0	0.0	2.51	2.5
10	10.0	1.0	1.0	0.0	1.0	0.0	5.0	5.0

Table 1: Process variables calculated from the process $G_{p1}(s)$ step response

	Frequency response		**Ziegler-Nichols**		**Cohen-Coon**		**Chien-Hrones-Reswick**	
T	K	T_i	K	T_i	K	T_i	K	T_i
0.1	0.257	0.373	0.09	3.3	0.173	0.287	0.06	0.1
0.2	0.275	0.425	0.18	3.3	0.263	0.413	0.12	0.2
0.5	0.365	0.633	0.45	3.3	0.533	0.735	0.3	0.5
1	0.571	1.067	0.9	3.3	0.983	1.138	0.6	1.0
2	1.041	2.026	1.8	3.3	1.883	1.658	1.2	2.0
5	2.509	5.0	4.5	3.3	4.583	2.354	3.0	5.0
10	5.035	10.0	9	3.3	9.083	2.755	6.0	10.0

Table 2: Controller parameters calculated by using different tuning methods for $G_{p1}(s)$

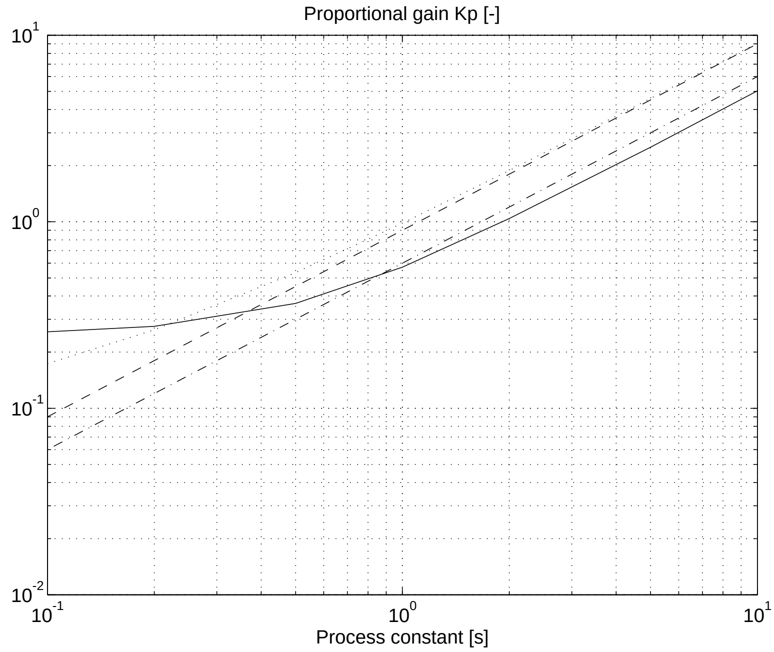

Fig. 6. Proportional gain (K) for process $G_{p1}(s)$;
__ FRM, -- ZN, ... CC, -.- CHR

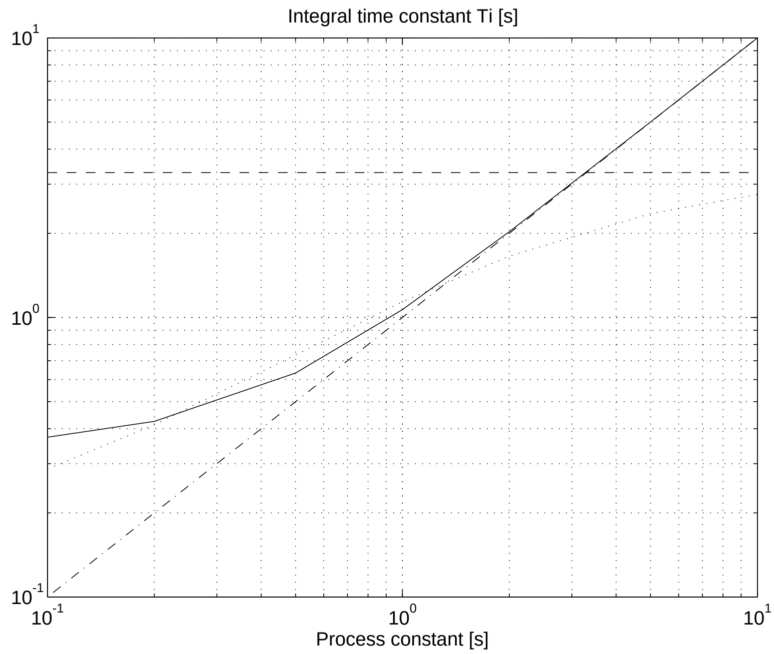

Fig. 7. Integral time constant (T_i) for process $G_{p1}(s)$;
__ FRM, -- ZN, ... CC, -.- CHR

Process 2:

$$G_{P2} = \frac{e^{-s}}{(1+sT)^2}$$

T	t_r	t_l	t_{ref}	y_{ref}	A_d	A_l	A_{rup}	A_{rdown}
0.1	0.272	1.028	1.1	0.264	1.09	0.001	0.07	0.037
0.2	0.544	1.056	1.2	0.264	1.18	0.002	0.147	0.074
0.5	1.359	1.141	1.5	0.264	1.45	0.004	0.368	0.184
1	2.718	1.282	2.0	0.264	1.90	0.009	0.736	0.368
2	5.432	1.563	3.0	0.264	2.79	0.018	1.469	0.737
5	13.59	2.409	6.0	0.264	5.48	0.044	3.678	1.84
10	27.17	3.817	11.0	0.264	9.97	0.087	7.352	3.684

Table 3: Process variables calculated from the process $G_{p2}(s)$ step response

	Frequency response		**Ziegler-Nichols**		**Cohen-Coon**		**Chien-Hrones-Reswick**	
T	K	T_i	K	T_i	K	T_i	K	T_i
0.1	0.262	0.412	0.238	3.39	0.322	0.502	0.159	0.272
0.2	0.285	0.508	0.464	3.48	0.547	0.791	0.309	0.544
0.5	0.371	0.852	1.07	3.77	1.155	1.439	0.715	1.359
1	0.49	1.485	1.91	4.23	1.991	2.18	1.272	2.72
2	0.636	2.798	3.13	5.16	3.21	3.27	2.09	5.43
5	0.804	6.784	5.08	7.95	5.16	5.86	3.38	13.6
10	0.893	13.46	6.41	12.60	6.49	9.83	4.27	27.2

Table 4: Controller parameters calculated by using different tuning methods for $G_{p2}(s)$

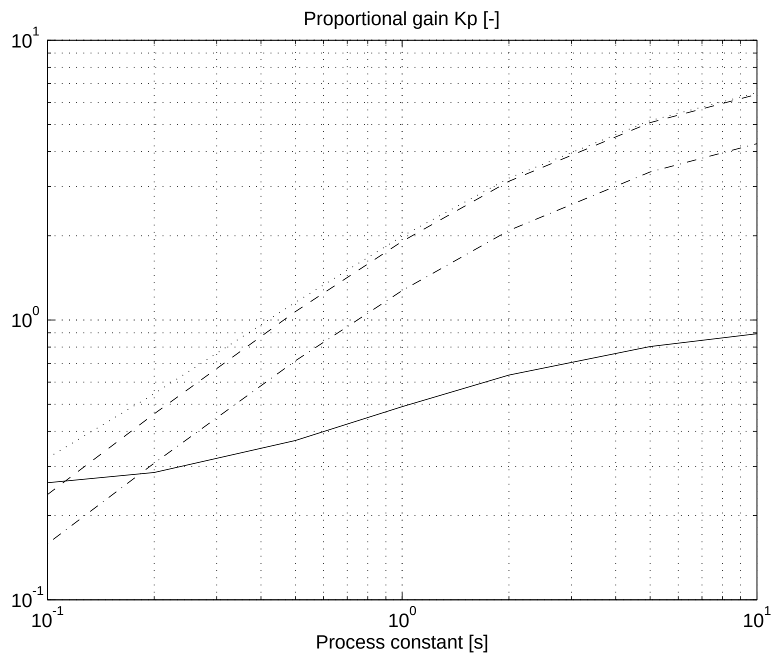

Fig. 8. Proportional gain (K) for process $G_{p2}(s)$;
__ FRM, -- ZN, ... CC, -.- CHR

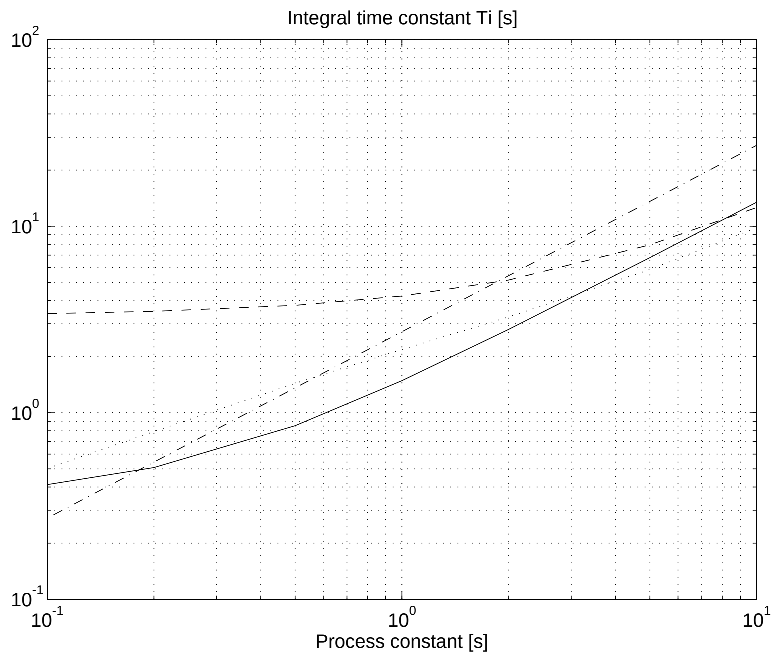

Fig. 9. Integral time constant (T_i) for process $G_{p2}(s)$;
__ FRM, -- ZN, ... CC, -.- CHR

Process 3:

$$G_{P3} = \frac{1}{(1+sT)(1+s)}$$

T	t_r	t_l	t_{ref}	y_{ref}	A_d	A_l	A_{rup}	A_{rdown}
0.1	1.29	0.064	0.256	0.149	0.24	0.001	0.468	0.391
0.2	1.5	0.107	0.408	0.201	0.38	0.002	0.477	0.348
0.5	2.0	0.193	0.698	0.251	0.63	0.006	0.559	0.313
1	2.72	0.282	1.0	0.264	0.90	0.009	0.736	0.368
2	4.0	0.386	1.39	0.251	1.25	0.011	1.122	0.625
5	7.47	0.535	2.02	0.201	1.86	0.012	2.399	1.74
10	12.88	0.643	2.56	0.148	2.41	0.011	4.668	3.925

Table 5: Process variables calculated from the process $G_{p3}(s)$ step response

	Frequency response		**Ziegler-Nichols**		**Cohen-Coon**		**Chien-Hrones-Reswick**	
T	K	T_i	K	T_i	K	T_i	K	T_i
0.1	5.04	1.0	18.15	0.211	18.24	0.193	12.1	1.291
0.2	2.61	1.007	12.57	0.353	12.66	0.31	8.38	1.495
0.5	1.25	1.007	9.33	0.637	9.41	0.535	6.22	2.0
1	1	1.333	8.67	0.931	8.76	0.772	5.78	2.72
2	1.25	2.143	9.33	1.274	9.41	1.070	6.22	4.0
5	2.61	5.033	12.57	1.766	12.66	1.55	8.38	7.47
10	5.04	10.0	18.03	2.12	18.11	1.94	12.02	12.88

Table 6: Controller parameters calculated by using different tuning methods for $G_{p3}(s)$

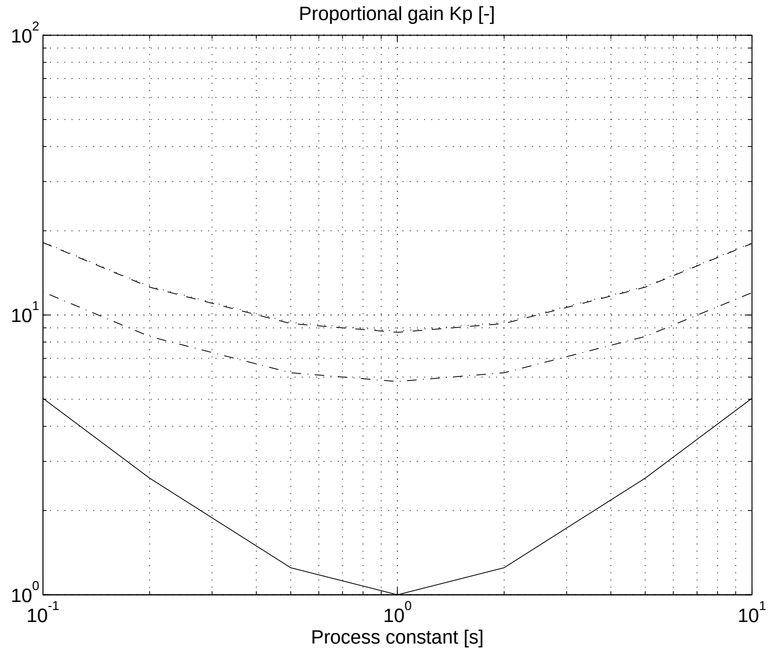

Fig. 10. Proportional gain (K) for process $G_{p3}(s)$;
__ FRM, -- ZN, ... CC, -.- CHR

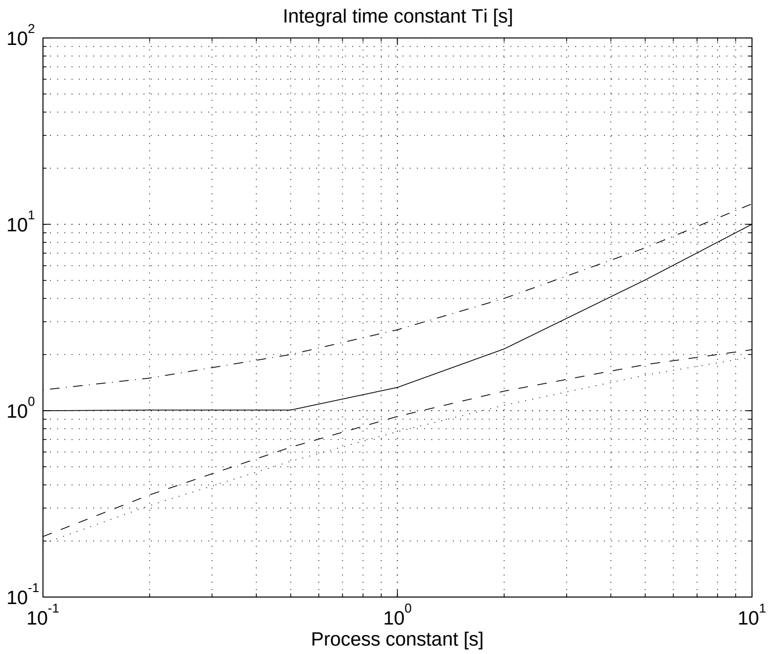

Fig. 11. Integral time constant (T_i) for process $G_{p3}(s)$;
__ FRM, -- ZN, ... CC, -.- CHR

Process 4:

$$G_{P4} = \frac{1}{(1+sT)^2(1+s)^2}$$

T	t_r	t_l	t_{ref}	y_{ref}	A_d	A_l	A_{rup}	A_{rdown}
0.1	2.75	0.473	1.23	0.275	1.11	0.011	0.722	0.364
0.2	2.85	0.637	1.48	0.295	1.34	0.016	0.708	0.352
0.5	3.39	1.005	2.15	0.338	1.93	0.028	0.742	0.329
1	4.46	1.426	3.0	0.353	2.68	0.042	0.935	0.385
2	6.78	2.01	4.3	0.338	3.86	0.056	1.485	0.658
5	14.27	3.187	7.38	0.294	6.69	0.078	3.556	1.758
10	27.49	4.725	12.23	0.273	11.1	0.110	7.267	3.638

Table 7: Process variables calculated from the process $G_{p4}(s)$ step response

	Frequency response		Ziegler-Nichols		Cohen-Coon		Chien-Hrones-Reswick	
T	K	T_i	K	T_i	K	T_i	K	T_i
0.1	0.81	1.359	5.23	1.561	5.31	1.16	3.49	2.75
0.2	0.695	1.396	4.03	2.10	4.11	1.451	2.69	2.85
0.5	0.545	1.565	3.03	3.32	3.12	2.08	2.02	3.39
1	0.5	2.0	2.82	4.71	2.9	2.87	1.878	4.46
2	0.545	3.131	3.03	6.63	3.12	4.16	2.02	6.78
5	0.696	6.978	4.03	10.52	4.11	7.26	2.69	14.27
10	0.808	13.60	5.24	15.59	5.32	11.59	3.49	27.49

Table 8: Controller parameters calculated by using different tuning methods for $G_{p4}(s)$

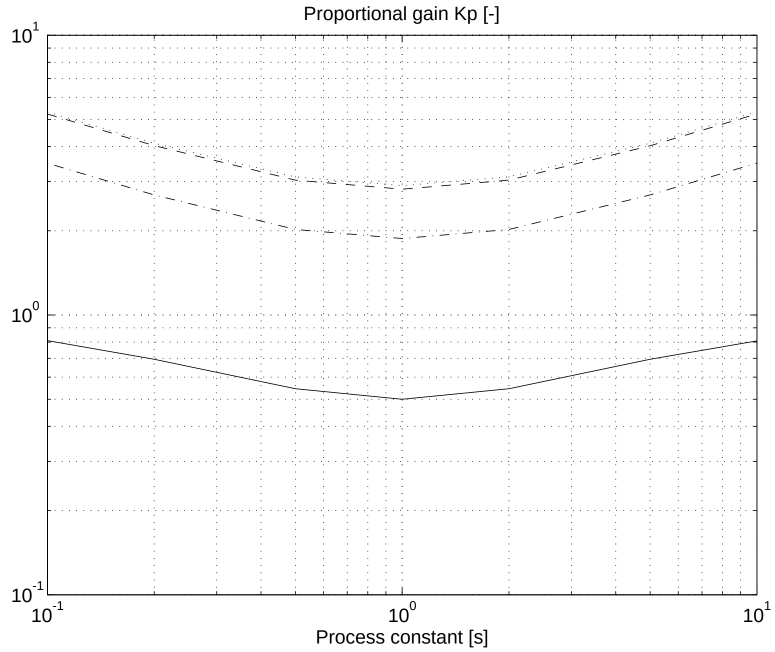

Fig. 12. Proportional gain (K) for process $G_{p4}(s)$;
__ FRM, -- ZN, ... CC, -.- CHR

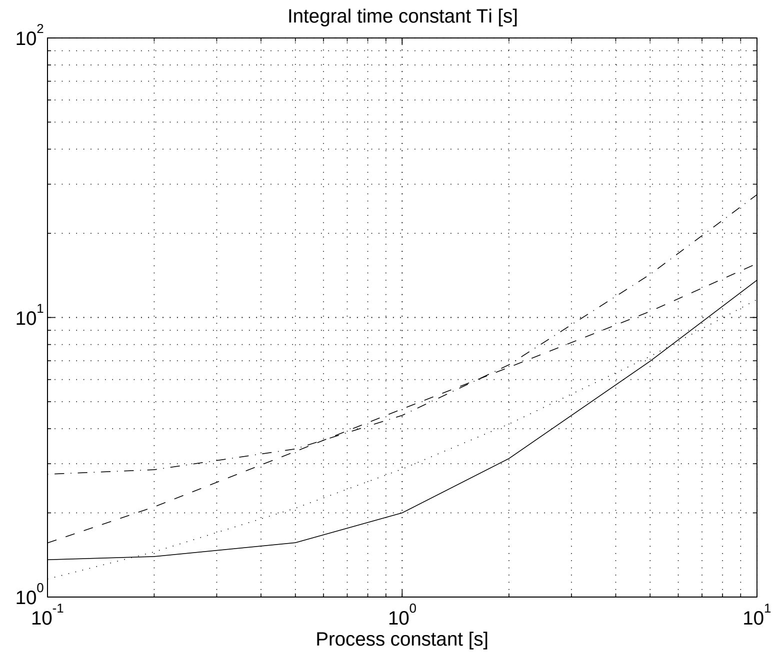

Fig. 13. Integral time constant (T_i) for process $G_{p4}(s)$;
__ FRM, -- ZN, ... CC, -.- CHR

Process 5:

$$G_{P5} = \frac{1}{(1+s)^n}$$

n	t_r	t_l	t_{ref}	y_{ref}	A_d	A_l	A_{rup}	A_{rdown}
2	2.714	0.283	1.0	0.27	0.907	0.009	0.724	0.369
3	3.692	0.806	2.0	0.32	1.774	0.025	0.853	0.373
4	4.46	1.426	3.0	0.355	2.688	0.042	0.927	0.385
5	5.12	2.1	4.0	0.377	3.608	0.058	0.993	0.398
6	5.697	2.81	5.0	0.39	4.528	0.073	1.06	0.412
7	6.224	3.55	6.0	0.396	5.438	0.088	1.137	0.426
8	6.71	4.307	7.0	0.405	6.373	0.101	1.188	0.439

Table 9: Process variables calculated from the process $G_{p5}(s)$ step response

	Frequency response		Ziegler-Nichols		Cohen-Coon		Chien-Hrones-Reswick	
n	K	T_i	K	T_i	K	T_i	K	T_i
2	1	1.33	8.63	0.934	8.71	0.774	5.75	2.71
3	0.625	1.66	4.12	2.66	4.21	1.849	2.75	3.69
4	0.5	2.0	2.81	4.71	2.90	2.87	1.877	4.46
5	0.437	2.33	2.19	6.93	2.28	3.81	1.463	5.12
6	0.4	2.66	1.825	9.27	1.908	4.69	1.216	5.70
7	0.375	3.0	1.578	11.72	1.661	5.52	1.052	6.22
8	0.357	3.33	1.402	14.21	1.486	6.30	0.935	6.71

Table 10: Controller parameters calculated by using different tuning methods for $G_{p5}(s)$

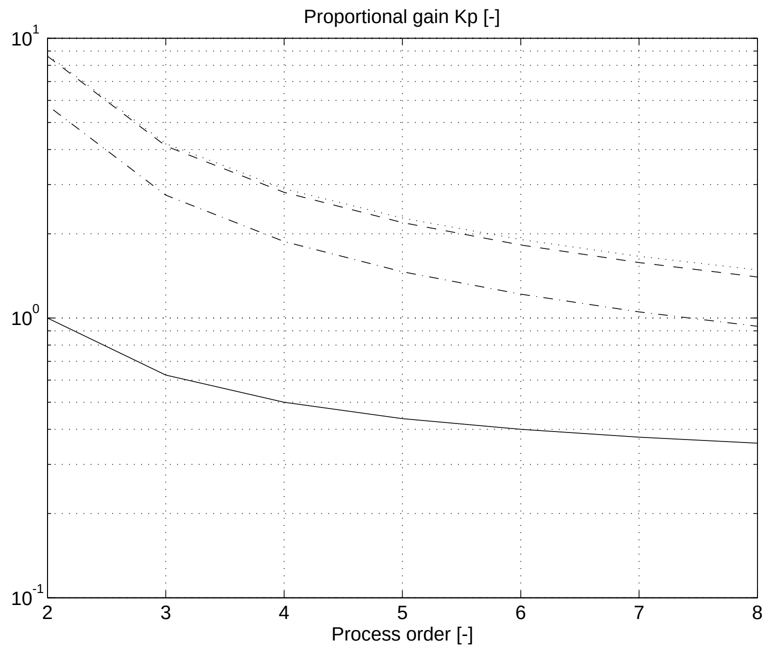

Fig. 14. Proportional gain (K) for process $G_{p5}(s)$;
__ FRM, -- ZN, ... CC, -.- CHR

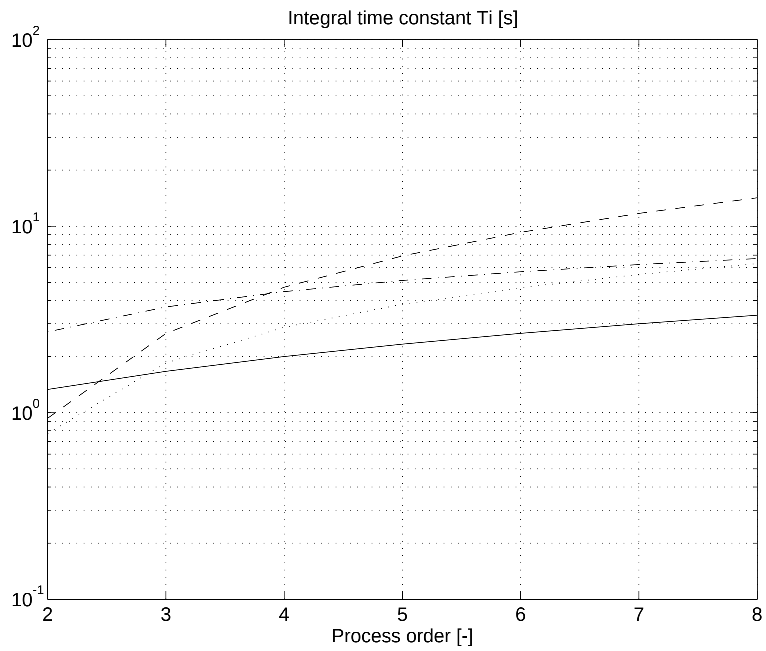

Fig. 15. Integral time constant (T_i) for process $G_{p5}(s)$;
__ FRM, -- ZN, ... CC, -.- CHR

Process 6:

$$G_{P6} = \frac{1}{(1+s)(1+sT)\left(1+sT^2\right)\left(1+sT^3\right)}$$

T	t_r	t_l	t_{ref}	y_{ref}	A_d	A_l	A_{rup}	A_{rdown}
0.2	1.50	0.154	0.468	0.21	0.43	0.003	0.469	0.346
0.3	1.71	0.25	0.664	0.243	0.61	0.005	0.489	0.32
0.4	1.93	0.366	0.883	0.268	0.81	0.008	0.516	0.302
0.5	2.19	0.504	1.16	0.3	1.05	0.012	0.535	0.291
0.6	2.50	0.66	1.46	0.321	1.32	0.017	0.576	0.286
0.7	2.87	0.832	1.8	0.338	1.61	0.023	0.629	0.29
0.8	3.31	1.02	2.16	0.345	1.93	0.029	0.712	0.306

Table 11: Process variables calculated from the process $G_{p6}(s)$ step response

	Frequency response		***Ziegler-Nichols***		***Cohen-Coon***		***Chien-Hrones-Reswick***	
T	***K***	T_i	***K***	T_i	***K***	T_i	***K***	T_i
0.2	2.106	1.01	8.77	0.508	8.85	0.422	5.84	1.5
0.3	1.318	1.03	6.14	0.825	6.22	0.638	4.09	1.705
0.4	0.948	1.06	4.74	1.208	4.82	0.874	3.16	1.928
0.5	0.751	1.13	3.91	1.663	3.99	1.137	2.60	2.19
0.6	0.634	1.22	3.40	2.18	3.49	1.422	2.27	2.50
0.7	0.564	1.34	3.10	2.75	3.19	1.735	2.07	2.87
0.8	0.525	1.51	2.92	3.37	3.01	2.08	1.949	3.31

Table 12: Controller parameters calculated by using different tuning methods for $G_{p6}(s)$

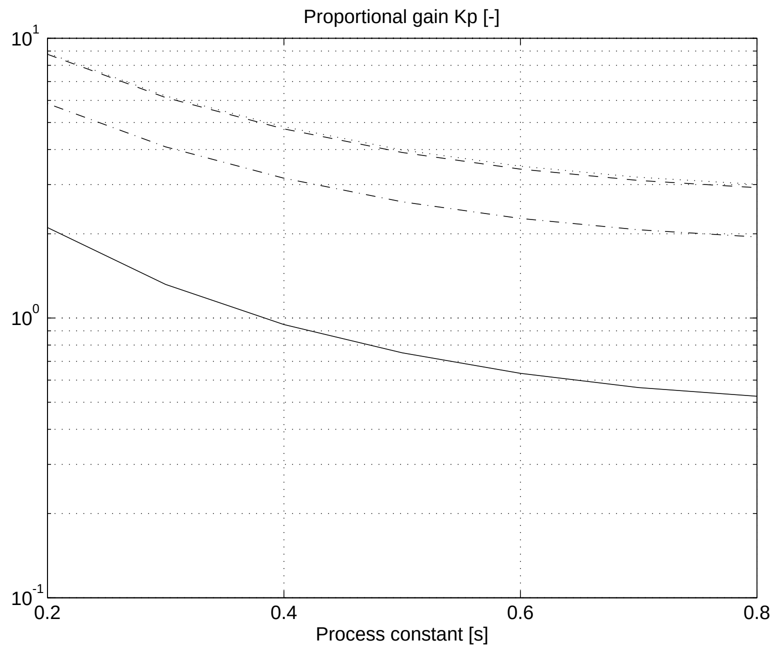

Fig. 16. Proportional gain (K) for process $G_{p6}(s)$;
__ FRM, -- ZN, ... CC, -.- CHR

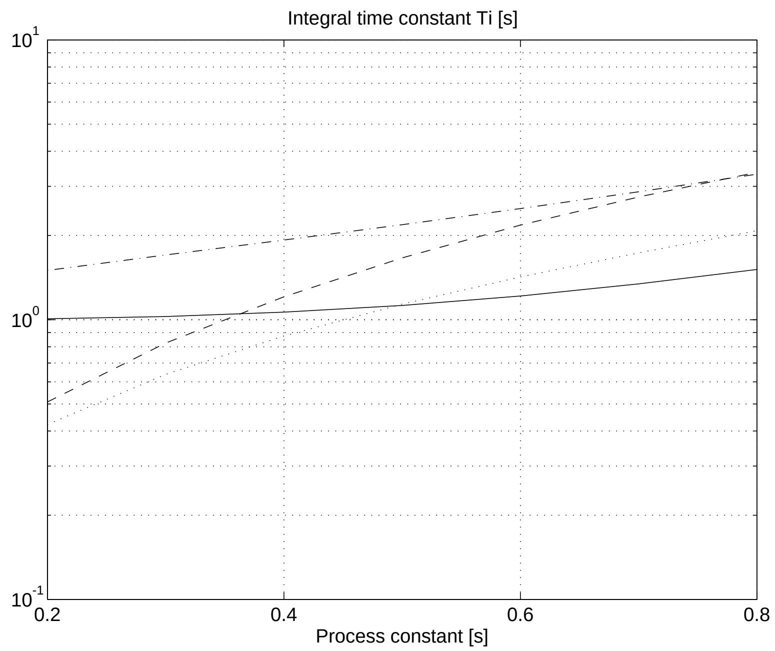

Fig. 17. Integral time constant (T_i) for process $G_{p6}(s)$;
__ FRM, -- ZN, ... CC, -.- CHR

Process 7:

$$G_{P7} = \frac{(1 - sT)}{(1 + s)^3}$$

T	t_r	t_l	t_{ref}	y_{ref}	A_d	A_l	A_{rup}	A_{rdown}
0.1	3.686	0.908	2.1	0.323	1.88	0.024	0.844	0.373
0.2	3.662	1.015	2.19	0.321	1.98	0.021	0.844	0.375
0.5	3.533	1.341	2.39	0.297	2.24	-0.005	0.873	0.387
1.0	3.236	1.854	2.62	0.237	2.64	-0.108	0.943	0.421
2.0	2.666	2.657	2.87	0.08	3.37	-0.504	1.128	0.507
5.0	1.665	3.905	3.14	-0.46	5.42	-2.458	1.773	0.805
10.0	1.008	4.678	3.27	-1.40	8.78	-6.49	2.896	1.325

Table 13: Process variables calculated from the process $G_{p7}(s)$ step response

	Frequency response		**Ziegler-Nichols**		**Cohen-Coon**		**Chien-Hrones-Reswick**	
T	K	T_i	K	T_i	K	T_i	K	T_i
0.1	0.594	1.68	3.65	3.00	3.74	2.00	2.44	3.69
0.2	0.566	1.70	3.25	3.35	3.33	2.15	2.16	3.66
0.5	0.49	1.73	2.37	4.43	2.45	2.52	1.58	3.53
1	0.4	1.78	1.571	6.12	1.654	2.87	1.05	3.24
2	0.289	1.83	0.903	8.77	0.986	3.03	0.602	2.67
5	0.156	1.91	0.384	12.89	0.467	2.59	0.256	1.665
10	0.088	1.95	0.194	15.44	0.277	2.02	0.129	1.008

Table 14: Controller parameters calculated by using different tuning methods for $G_{p7}(s)$

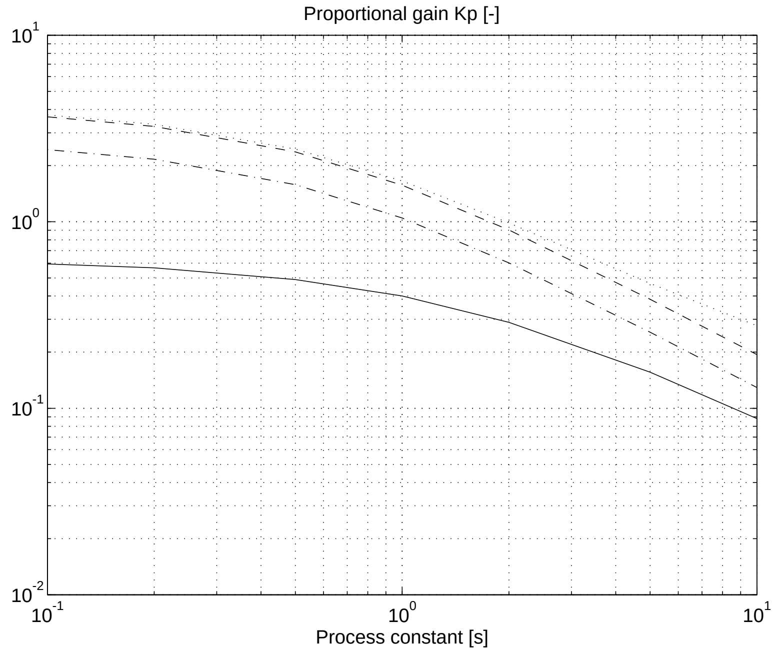

Fig. 18. Proportional gain (K) for process $G_{p7}(s)$;
__ FRM, -- ZN, ... CC, -.- CHR

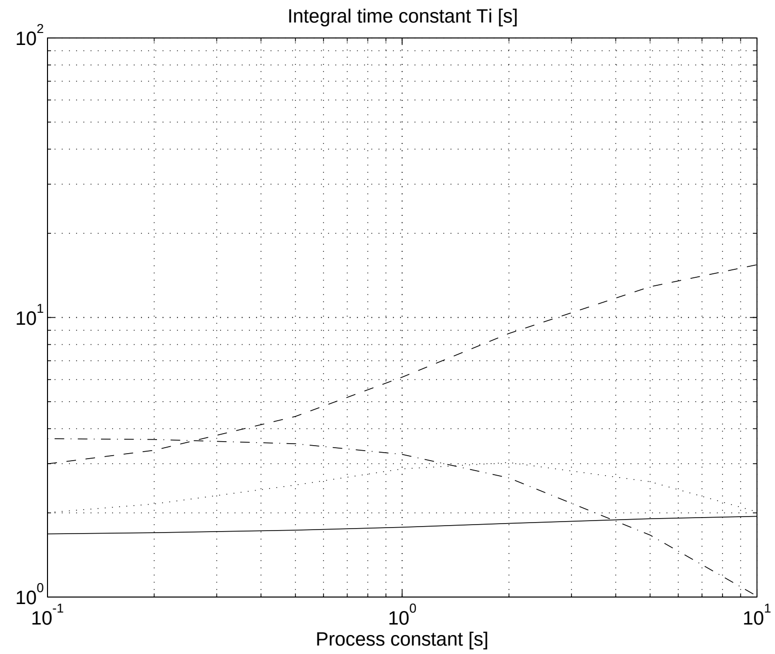

Fig. 19. Integral time constant (T_i) for process $G_{p7}(s)$;
__ FRM, -- ZN, ... CC, -.- CHR

Process 8:

$$G_{P8} = \frac{e^{-} \left(1 + sT\right)}{\left(1 + s\right)}$$

T	t_r	t_l	t_{ref}	y_{ref}	A_d	A_l	A_{rup}	A_{rdown}
0.1	2.703	1.186	1.9	0.264	1.80	0.008	0.732	0.37
0.2	2.646	1.104	1.76	0.248	1.67	0.005	0.748	0.378
0.3	2.53	1.04	1.58	0.213	1.52	0.002	0.784	0.395
0.4	2.326	1.007	1.34	0.143	1.32	0.0	0.854	0.43
0.5	2.0	1.0	1.02	0.0	1.02	0.0	0.98	0.5
0.7	1.437	1.0	1.02	0.0	1.02	0.0	0.698	0.582
1.0	1.01	1.0	1.018	0.0	1.02	0.0	0.487	0.495

Table 15: Process variables calculated from the process $G_{p8}(s)$ step response

	Frequency response		**Ziegler-Nichols**		**Cohen-Coon**		**Chien-Hrones-Reswick**	
T	***K***	T_i	***K***	T_i	***K***	T_i	***K***	T_i
0.1	0.51	1.47	2.05	3.91	2.13	2.09	1.368	2.70
0.2	0.53	1.44	2.16	3.64	2.24	1.989	1.438	2.65
0.3	0.552	1.42	2.19	3.43	2.27	1.886	1.46	2.53
0.4	0.573	1.39	2.08	3.32	2.16	1.785	1.386	2.33
0.5	0.59	1.35	1.8	3.3	1.883	1.658	1.2	2.0
0.7	0.616	1.27	1.29	3.3	1.377	1.40	0.862	1.437
1.0	0.571	1.07	0.909	3.3	0.992	1.145	0.606	1.01

Table 16: Controller parameters calculated by using different tuning methods for $G_{p8}(s)$

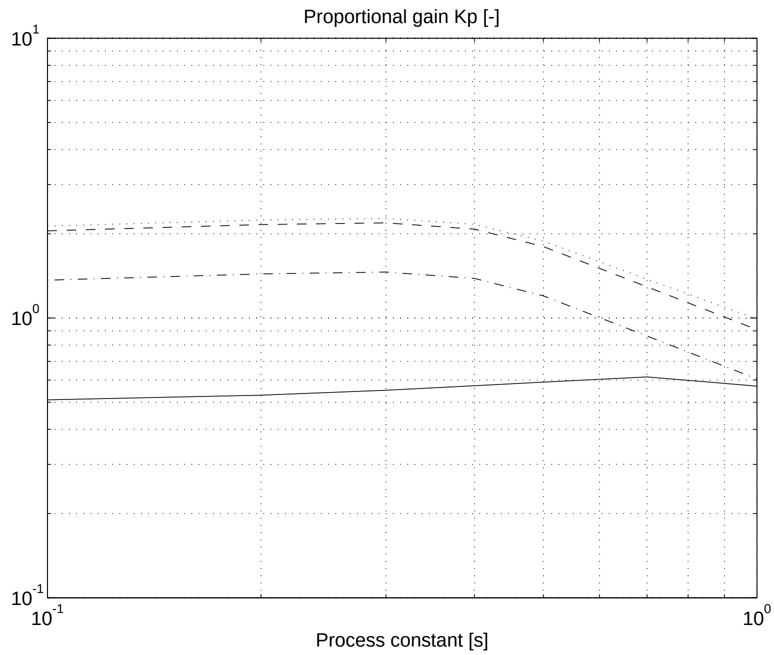

Fig. 20. Proportional gain (K) for process $G_{p8}(s)$;
__ FRM, -- ZN, ... CC, -.- CHR

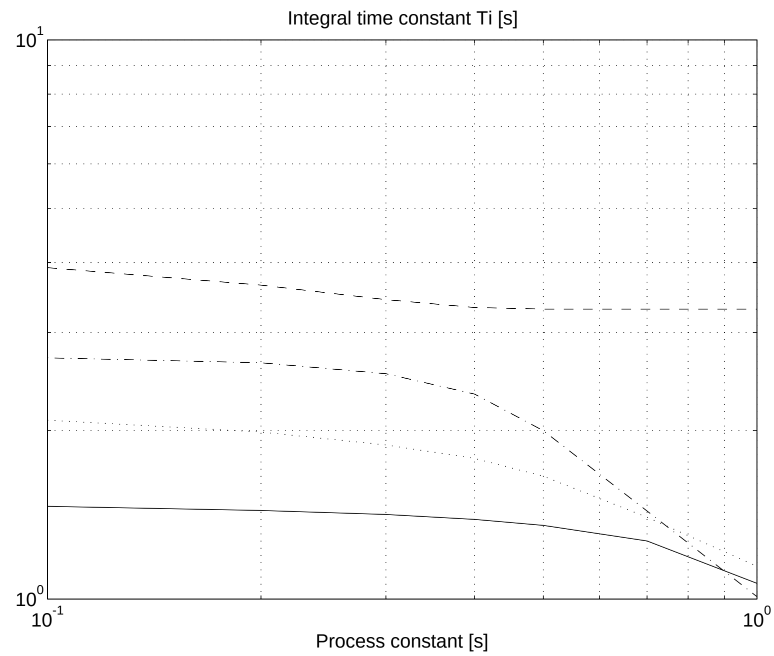

Fig. 21. Integral time constant (T_i) for process $G_{p8}(s)$;
__ FRM, -- ZN, ... CC, -.- CHR

Process 9:

$$G_{p9} = \frac{1}{(1+s)\big(1+s(1+i\alpha)\big)\big(1+s(1-i\alpha)\big)}$$

α	t_r	t_l	t_{ref}	y_{ref}	A_d	A_l	A_{rup}	A_{rdown}
0.1	3.692	0.81	2.038	0.333	1.81	0.025	0.822	0.369
0.2	3.692	0.82	2.039	0.330	1.81	0.026	0.828	0.36
0.3	3.693	0.836	2.08	0.337	1.84	0.026	0.812	0.344
0.4	3.696	0.858	2.12	0.342	1.88	0.028	0.801	0.322
0.5	3.701	0.885	2.16	0.345	1.91	0.029	0.795	0.293
0.7	3.721	0.954	2.32	0.367	2.04	0.033	0.745	0.218
1.0	3.786	1.084	2.595	0.40	2.25	0.041	0.683	0.065

Table 17: Process variables calculated from the process $G_{p9}(s)$ step response

	Frequency response		**Ziegler-Nichols**		**Cohen-Coon**		**Chien-Hrones-Reswick**	
T	***K***	T_i	***K***	T_i	***K***	T_i	***K***	T_i
0.1	0.62	1.66	4.10	2.67	4.19	1.855	2.73	3.69
0.2	0.607	1.64	4.05	2.71	4.14	1.871	2.70	3.69
0.3	0.585	1.62	3.98	2.76	4.06	1.896	2.65	3.69
0.4	0.553	1.58	3.88	2.83	3.96	1.931	2.58	3.70
0.5	0.514	1.52	3.76	2.92	3.85	1.972	2.51	3.70
0.7	0.42	1.37	3.51	3.15	3.59	2.08	2.34	3.72
1.0	0.25	1.0	3.14	3.58	3.23	2.27	2.10	3.79

Table 18: Controller parameters calculated by using different tuning methods for $G_{p9}(s)$

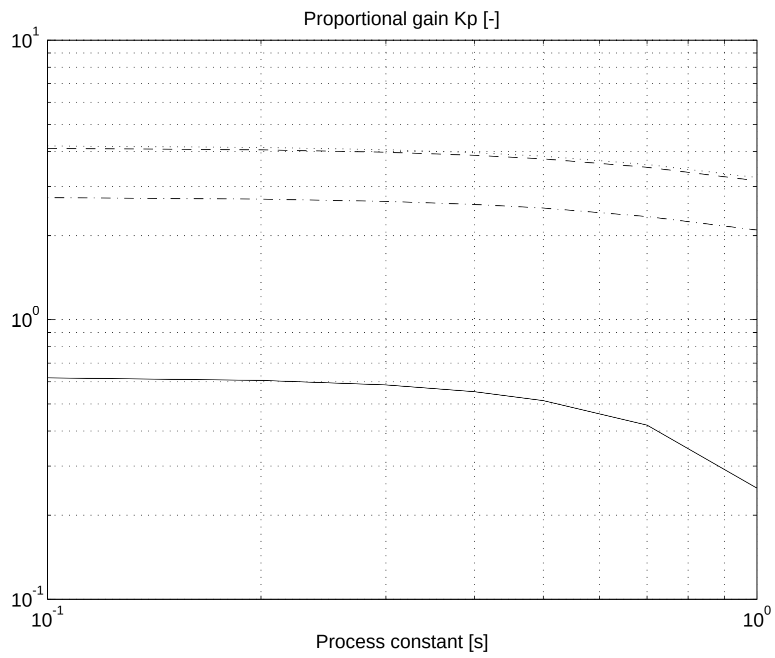

Fig. 22. Proportional gain (K) for process $G_{p9}(s)$;
__ FRM, -- ZN, ... CC, -.- CHR

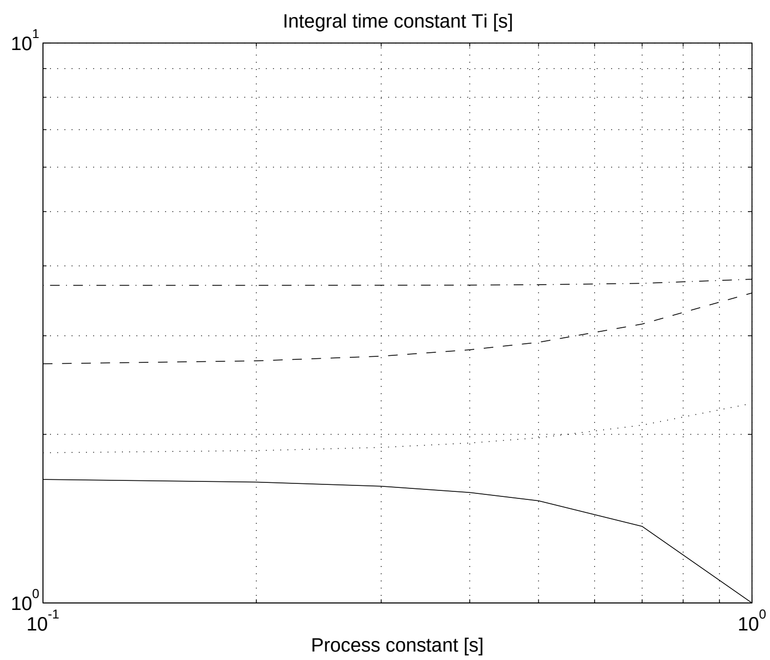

Fig. 23. Integral time constant (T_i) for process $G_{p9}(s)$;
__ FRM, -- ZN, ... CC, -.- CHR

5. Closed-loop experiments

Several closed-loop experiments were performed, where tracking and control performances were being tested. The set-point changed from 0 to 1 at time origin, and the disturbance unity step change at process input appeared at the half-time of each experiment. Three different process constants were chosen for each process model (marked with shadowed rows in Tables 1 to 18).

Time responses were obtained by running the simulation scheme in SIMULINK, shown in Fig. 24.

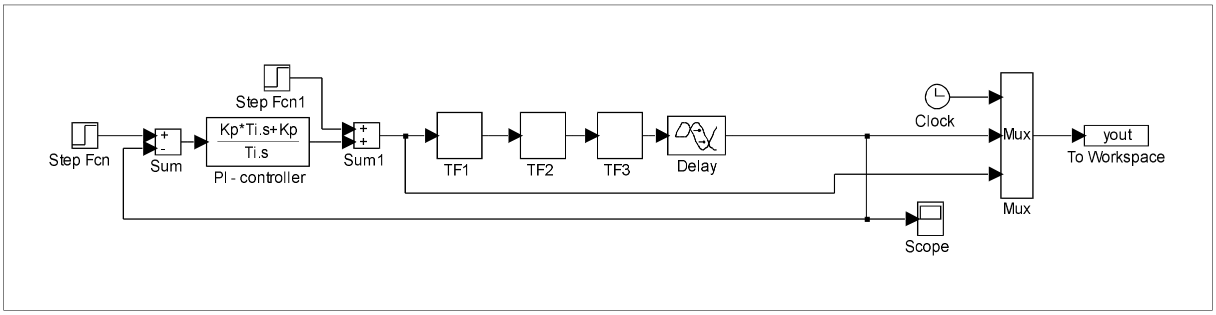

Fig. 24. The SIMULINK scheme for testing the closed-loop time response.

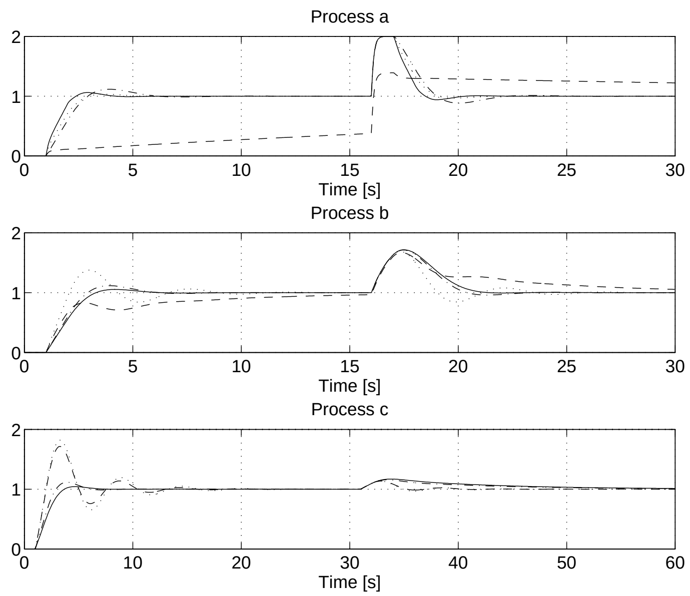

Fig. 25. Closed-loop process response on the reference (w) and the disturbance (d) step change for $G_{P1}(s)$;
a) T=0.1, b) T=1, c) T=10; __ FRM, -- ZN, ... CC, -.- CHR

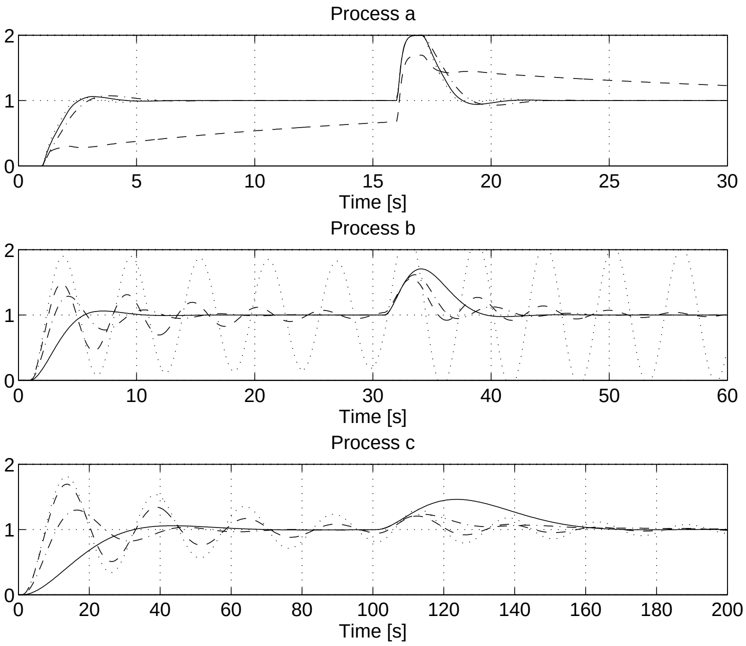

Fig. 26. Closed-loop process response on the reference (w) and the disturbance (d) step change for $G_{P2}(s)$; a) T=0.1, b) T=1, c) T=10; __ FRM, -- ZN, ... CC, -.- CHR

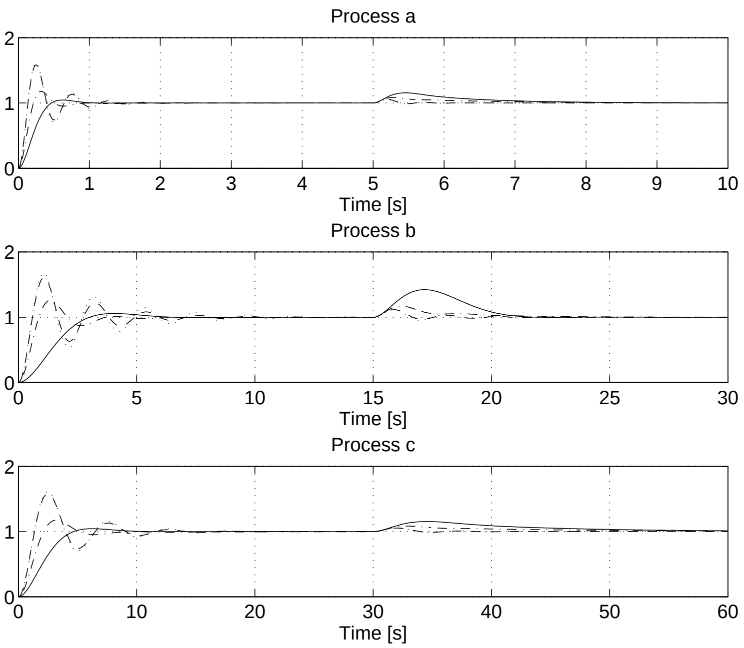

Fig. 27. Closed-loop process response on the reference (w) and the disturbance (d) step change for $G_{P3}(s)$; a) T=0.1, b) T=1, c) T=10; __ FRM, -- ZN, ... CC, -.- CHR

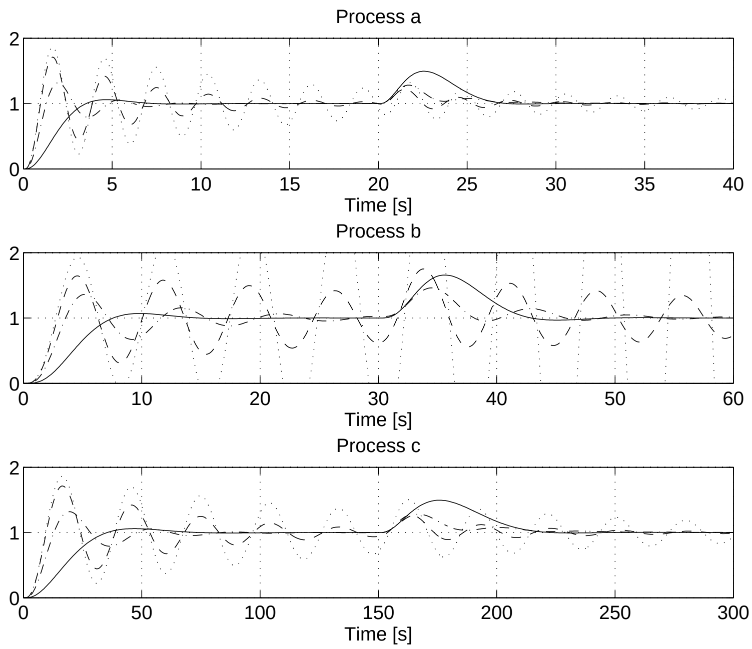

Fig. 28. Closed-loop process response on the reference (w) and the disturbance (d) step change for $G_{P4}(s)$; a) T=0.1, b) T=1, c) T=10; __ FRM, -- ZN, ... CC, -.- CHR

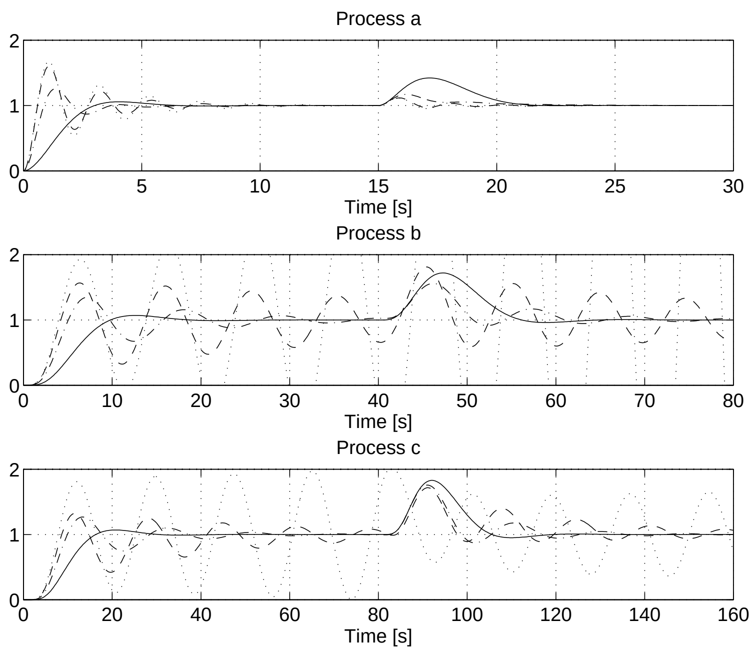

Fig. 29. Closed-loop process response on the reference (w) and the disturbance (d) step change for $G_{P5}(s)$; a) n=2, b) n=5, c) n=8; __ FRM, -- ZN, ... CC, -.- CHR

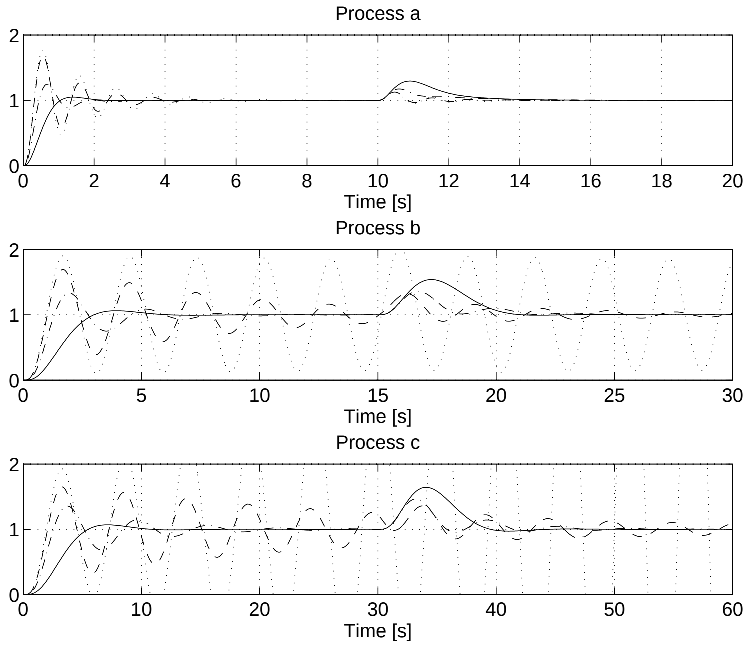

Fig. 30. Closed-loop process response on the reference (w) and the disturbance (d) step change for $G_{P6}(s)$; a) T=0.2, b) T=0.5, c) T=0.8; __ FRM, -- ZN, ... CC, -.- CHR

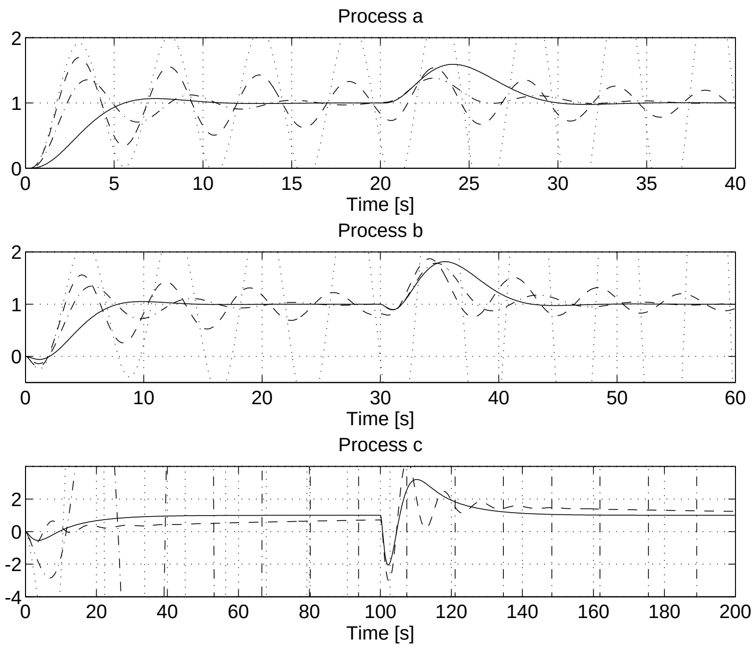

Fig. 31. Closed-loop process response on the reference (w) and the disturbance (d) step change for $G_{P7}(s)$; a) T=0.1, b) T=1, c) T=10; __ FRM, -- ZN, ... CC, -.- CHR

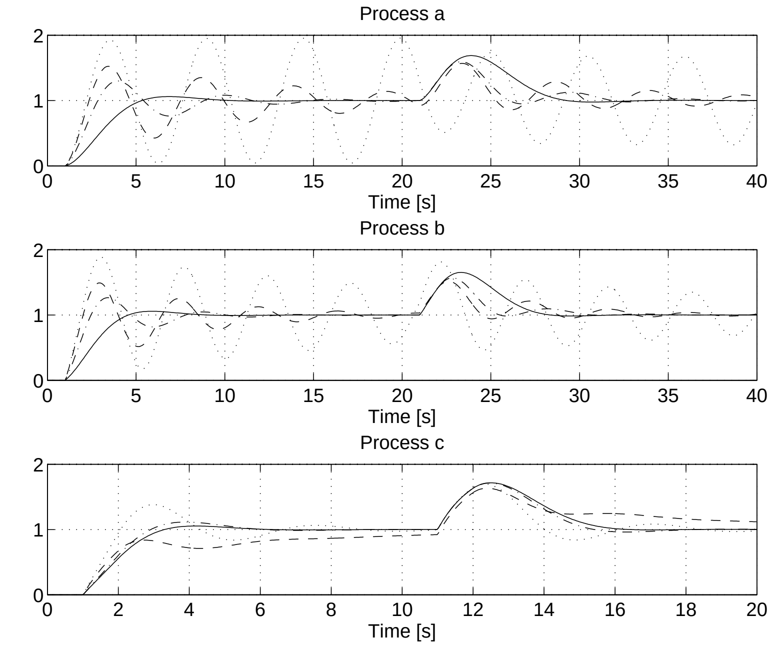

Fig. 32. Closed-loop process response on the reference (w) and the disturbance (d) step change for $G_{P8}(s)$; a) T=0.1, b) T=0.4, c) T=1; __ FRM, -- ZN, ... CC, -.- CHR

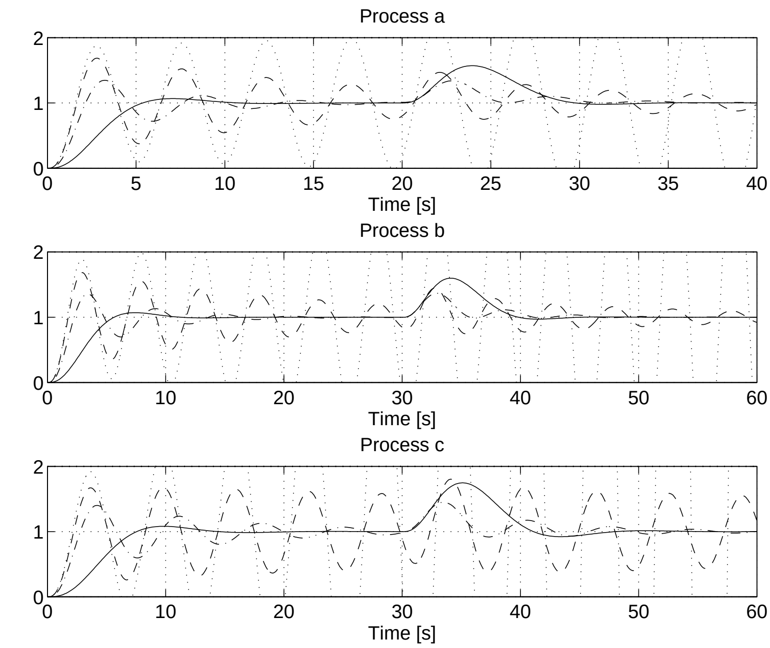

Fig. 33. Closed-loop process response on the reference (w) and the disturbance (d) step change for $G_{P9}(s)$; a) α=0.1, b) α=0.4, c) α=1; __ FRM, -- ZN, ... CC, -.- CHR

6. Conclusions

The experimental results showed that the first three methods, which are based on the process reaction curve, become unstable at least once. Concerning stability, the best among the three of them was Chien-Hrones-Reswick (CHR) method, which was unstable only when applying the process $G_{P7}(s)$ with T=10s (highly nonminimal phase process).

The frequency response method (FRM) always gives a stable closed-loop response. Moreover, the closed-loop responses have similar shapes.

The drawback of the latter method lies in the fact that the complete process model has to be known a-priori.

Our further work will therefore be based on finding such PI controller parameters as will be close to the results given by FRM, but will need less a-priori knowledge about the process. If possible, the parameter calculations should be based only on the process step response.

Appendix

The MATLAB procedure FINDKPTI.M:

```
% FINDX.M calculates desired Kp and Ti from the process transfer function
% such that the open-loop Nyquist curve has desired shape (close to the
% vertical line Re=-1/2). It assures the aperiodic response in time domain.
% The exception are the processes with strong zeros...

Tiplus = 1000;
Timinus = 0.01;

Kp=1;
Ti=0.1;
Td=0;
Tiold = 0.1;

poleorig=0;
zeroorig=0;

imag = sqrt(-1);
Points = 250;
w = logspace (-3,3,Points)+0.001;

rez = [];

w=w';
up =1;

while (up == 1)

  Tiold = Ti;

  rez = [rez; Kp Ti];

  stevec = conv([Kp*Ti Kp],[Tz1*Tz2*Tz3 Tz1*Tz2+Tz2*Tz3+Tz1*Tz3 Tz1+Tz2+Tz3 1]);
  if (zeroorig == 1),
    stevec = conv([Kp*Ti Kp 0],[Tz1*Tz2*Tz3 Tz1*Tz2+Tz2*Tz3+Tz1*Tz3 Tz1+Tz2+Tz3 1]);
  end
  imenovalec = conv([Ti 0],[Tp1*Tp2*Tp3 Tp1*Tp2+Tp2*Tp3+Tp1*Tp3 Tp1+Tp2+Tp3 1]);
  imenovalec = conv(imenovalec,[Tp4*Tp5*Tp6 Tp4*Tp5+Tp5*Tp6+Tp4*Tp6 Tp4+Tp5+Tp6 1]);
  imenovalec = conv(imenovalec,[Tp7*Tp8 Tp7+Tp8 1]);
  if (poleorig == 1),
    imenovalec = conv([Ti 0 0],[Tp1*Tp2*Tp3 Tp1*Tp2+Tp2*Tp3+Tp1*Tp3 Tp1+Tp2+Tp3 1]);
    imenovalec = conv(imenovalec,[Tp4*Tp5*Tp6 Tp4*Tp5+Tp5*Tp6+Tp4*Tp6 Tp4+Tp5+Tp6 1]);
    imenovalec = conv(imenovalec,[Tp7*Tp8 Tp7+Tp8 1]);
  end
  if (n ~= 1),
    imenovalec = conv([Ti 0],[Tp1 1]);
    for i=2:n,
      imenovalec = conv(imenovalec,[Tp1 1]);
    end
  end

  [re,im] = nyquist(stevec,imenovalec,w);
  points = re+imag*im;
  ampl=abs(points);
  phase=angle(points);
  phase=phase-w*Tdelay;
  re=ampl.*cos(phase);
  im=ampl.*sin(phase);
  points = re+imag*im;

  i = 1;
  while ((im(i+1) < 0) & (i < (Points-1)))
    i = i+1;
  end
  rex2 = re(i)+(re(i+1)-re(i))*(im(i))/(im(i)-im(i+1));
  w01 = w(i)+(w(i+1)-w(i))*(im(i))/(im(i)-im(i+1));
  A01 = -rex2;

  i = 2;
  sumi = 0;
```

```
  npt = 0;
  while ((im(i) < 0) & (i < Points))
    i = i+1;
  end

  sumi = 1+sign(re(1)-re(1:i-1));
  sumi(1)=0;
  sumi = mean(sumi);

  if (sumi > 0)
    up = 0;
  else
    Ti = Ti*2;
  end

end

Tiplus = Ti;
Timinus = Ti/2;
Ti = (Tiplus+Timinus)/2;

for m=1:10,

  rez = [rez; Kp Ti];

  stevec = conv([Kp*Ti Kp],[Tz1*Tz2*Tz3 Tz1*Tz2+Tz2*Tz3+Tz1*Tz3 Tz1+Tz2+Tz3 1]);
  if (zeroorig == 1),
    stevec = conv([Kp*Ti Kp 0],[Tz1*Tz2*Tz3 Tz1*Tz2+Tz2*Tz3+Tz1*Tz3 Tz1+Tz2+Tz3 1]);
  end
  imenovalec = conv([Ti 0],[Tp1*Tp2*Tp3 Tp1*Tp2+Tp2*Tp3+Tp1*Tp3 Tp1+Tp2+Tp3 1]);
  imenovalec = conv(imenovalec,[Tp4*Tp5*Tp6 Tp4*Tp5+Tp5*Tp6+Tp4*Tp6 Tp4+Tp5+Tp6 1]);
  imenovalec = conv(imenovalec,[Tp7*Tp8 Tp7+Tp8 1]);
  if (poleorig == 1),
    imenovalec = conv([Ti 0 0],[Tp1*Tp2*Tp3 Tp1*Tp2+Tp2*Tp3+Tp1*Tp3 Tp1+Tp2+Tp3 1]);
    imenovalec = conv(imenovalec,[Tp4*Tp5*Tp6 Tp4*Tp5+Tp5*Tp6+Tp4*Tp6 Tp4+Tp5+Tp6 1]);
    imenovalec = conv(imenovalec,[Tp7*Tp8 Tp7+Tp8 1]);
  end
  if (n ~= 1),
    imenovalec = conv([Ti 0],[Tp1 1]);
    for i=2:n,
      imenovalec = conv(imenovalec,[Tp1 1]);
    end
  end

  [re,im] = nyquist(stevec,imenovalec,w);
  points = re+imag*im;
  ampl=abs(points);
  phase=angle(points);
  phase=phase-w*Tdelay;
  re=ampl.*cos(phase);
  im=ampl.*sin(phase);
  points = re+imag*im;

  i = 1;
  while ((im(i+1) < 0) & (i < (Points-1)))
    i = i+1;
  end
  rex2 = re(i)+(re(i+1)-re(i))*(im(i))/(im(i)-im(i+1));
  w01 = w(i)+(w(i+1)-w(i))*(im(i))/(im(i)-im(i+1));
  A01 = -rex2;


  i = 2;
  sumi = 0;
  npt = 0;
  while ((im(i) < 0) & (i < Points))
    i = i+1;
  end

  sumi = 1+sign(re(1)-re(1:i-1));
  sumi(1)=0;
  sumi = mean(sumi);

  if (sumi > 0)
    Tiplus = Ti;
    Ti = (Ti+Timinus)/2;
  else
    Timinus = Ti;
    Ti = (Ti+Tiplus)/2;
  end

end
```

```
Kp = -0.5/re(1);

i = 2;
sumi = 0;
npt = 0;
while ((im(i) < 0) & (i < Points))
  i = i+1;
end
sumi = 1+sign(re(1)-re(1:i-1));
sumi(1)=0;
sumi = mean(sumi);

wu=w(i);

% x - the ration between Ti and the period of the ultimate frequency

x=Ti*wu;

plot(points)
axis([-4 4 -4 4]);
grid
pause(0.2);
```

The MATLAB procedure FINDALL.M:

```
% [tr,tl,tref,yref,K1,Ad,Al,Arup,Ardown,ratmin,ratar,rat,Kp,Ti] =
% findall(yout,Tdelay,Tp1,Tp2,Tp3,Tp4,Tp5,Tp6,Tp7,Tp8,Tz1,Tz2,Tz3);
%
% Function FINDALL.M finds different process values from process
% step response. Output values are:
%
%      tr - process rise time
%      tl - process lag time
%      tref - replection point time
%      yref - reflection point value
%      K1 - the area, surronded by process
%           input and output step response
%      Ad - the area surrounded by process step response
%           and line t=tref
%      Al - the area surrounded by process step response
%           and the tangent on the reflection point
%      Arup - the are surrounded by tangent on the refl.
%             point and t=tref
%      Ardown - the area surrounded by process step response
%               and tangent on the reflection point
%      ratmin - minimal value of the variable rat (is appr. valid for
%               the first order process with a delay)
%      ratar - the ratio between areas Ardown and Arup (ratar=1 for
%              the first order process with a delay)
%      rat - the actual ratio for the given process. It is a result
%            of the criterion based on the Nyquist curve shape.
%      Kp - is calculated as Kp = 0.5/rat to reach desired shape of the
%           Nyuist curve
%      Ti - is calculated as Ti = K1/(1+rat) to reach desired shape of
%           the Nyquist curve
%      yout - the process response; 1. column = time, 2. column = process output
%      Tdelay - process delay
%      Tp1..Tp8 - poles of the process: .../((1+s*Tp1)*(1+s*Tp2)*...*(1+s*Tp8))
%      Tz1..Tz3 - zeros of the process: ((1+s*Tz1)*(1+s*Tz2)*(1+s*Tz3))/...

function [tr,tl,tref,yref,K1,Ad,Al,Arup,Ardown,ratmin,ratar,rat,Kp,Ti] =
findall(yout,Tdelay,Tp1,Tp2,Tp3,Tp4,Tp5,Tp6,Tp7,Tp8,Tz1,Tz2,Tz3);

x = yout(:,1);
y = yout(:,2);

dim = max(size(x));
dx=diff(x);
dx = [dx(1);dx];
dy=diff(y);
dy = [dy(1);dy];
dydx = dy./dx;
[a,i]=max(dydx);
```

```
tr = (y(dim)-y(1))/a;
tl = x(i)-y(i)/a;
iref = i;

tref = x(i);
yref = y(i);
Ad = yout(i,6);

ratmin=2/sqrt(1+4*(tr/tl)^2);

K1 = Tdelay+Tp1+Tp2+Tp3+Tp4+Tp5+Tp6+Tp7+Tp8-Tz1-Tz2-Tz3;
Al = tref-Ad-yref*(tref-tl)/2;
Arup = (1-yref)*(tl+tr-tref)/2;
Ardown = SumA-Ad-Arup;

ratar = Ardown/Arup;

n=1;
findkpti;

rat = 0.5/Kp;
```

References

K. J. Åström and T. Hägglund, "**New Tuning Methods for PID Controllers**", ECC 1995.

P. Boucher and Y. Tanguy, "**Une nouvelle méthode de réglage des régulateurs PI ou PID**", Automatisme, May, 1976.

J. J. DiStefano, "**Feedback and Control Systems**", 2nd edition, Schaum's outline series, McGraw-Hill, 1990.

A. Haalman, "**The adjustment of continuous controllers**", Review A, VIII, 4, pp. 181-199, 1966.

C. C. Hang, K. J. Åström and W. K. Ho, "**Refinements of the Ziegler-Nichols tuning formula**", IEE Proceedings, D, Vol. 138, No. 2, pp. 111-118, 1991.

C. C. Hang and L. S. Cao, "**Improvement of Transient Response by Means of Setpoint Weighting**", 12th World Congress IFAC, Sydney, Proceedings, Vol. 3, pp. 69-72, 1993.

R. Hanus, "**Determination of controllers parameters in the frequency domain**", Journal A, Vol. XVI, No. 3, 1975.

P. Šega, "**Control and Systems (in Slovene)**", FER, Ljubljana, 1991.

D. Vrančić and Y. Peng, "**Amplitude and phase margin detection with on-line PID controller**", J. Stefan Institute, Report DP-7054, (1994).

D. Vrančić, J. Petrovčič, D. Juričić, "**PID parameters setting for the 2nd order processes (in Slovene)**", J. Stefan Institute, Report DP-6782, (1993).