# Experimental Verification of Design Methods for Conventional PI/PID Controllers

RADEK MATUŠŮ, ROMAN PROKOP
Department of Automation and Control Engineering
Faculty of Applied Informatics
Tomas Bata University in Zlín
Nad Stráněmi 4511, 76005 Zlín
CZECH REPUBLIC
{rmatusu; prokop}@fai.utb.cz

*Abstract:* - The paper presents an experimental verification of conventional PI and PID controller design methods. Nine different synthesis techniques were used to control of airflow speed in laboratory model of hot-air tunnel, most of them for both PI and PID version. In the first instance the controlled plant was identified in order to obtain a linear mathematical model and then the controllers were designed using studied methods. The controllers were tested in simulations and subsequently implemented under real conditions. The obtained control results have been compared and analyzed.

*Key-Words:* - Control Design Methods, PID Control, Linear Control, Hot-Air Tunnel, Identification

## 1 Introduction

Since the pioneering work of Ziegler and Nichols in 1940's [1], many researchers and control engineers have tried to find an appropriate way of setting the Proportional (P), Integral (I) and Derivative (D) terms in conventional controllers. Naturally, this long-time interest has brought many design approaches, tuning rules, formulae and recommendations [2], [3], [4]. However, the PID control is not only the matter of history because more than 95% of contemporary practical industrial control applications exploit PI(D) controllers. Unfortunately, many of them are not tuned properly. So, the question of PID control design is still very topical [5], [6], [7], [8], [9].

The paper does not intend to bring any new method. Its main aim is to experimentally verify the nine selected PI(D) control design methods and compare their simulative and real results. The set of tested techniques include representatives of the "old school" as well as some newer approaches. The list covers:

- Chien-Hrones-Reswick Method
- Cohen-Coon Method
- Step Response Method
- Whiteley Standard Forms Method
- Naslin Method
- Fruehauf Method
- Balanced Tuning of PI Controllers
- Desired Model Method
- Polynomial Synthesis

All the methods were tested during control of airflow speed in laboratory model of hot-air tunnel. The control outputs have been compared and discussed.

## 2 Description of Hot-Air Tunnel and Used Equipment

The controlled plant, used for experiments, has been represented by laboratory model of hot-air tunnel constructed in VŠB – Technical University of Ostrava [10]. Generally, this object can be seen as multi-input multi-output (MIMO) system, however the tests have been done on a selected single-input single-output (SISO) loop. The model is composed of the bulb, primary and secondary ventilator and an array of sensors covered by tunnel. The bulb is powered by controllable source of voltage and serves as the source of light and heat energy while the purpose of ventilators is to ensure the flow of air inside the tunnel. All components are connected to the electronic circuits which adjust signals into the voltage levels suitable for CTRL 51 unit. Finally, this control unit is connected with the PC via serial link RS232. The real visual appearance is shown in Fig. 1 while Fig. 2 presents the simplified diagram (only by reason of convenient model orientation and "nicer" illustration, the secondary ventilator is formally depicted on the opposite side than in the real case).

Fig. 1: Model of Hot-Air Tunnel Connected to PC via CTRL 51 Unit

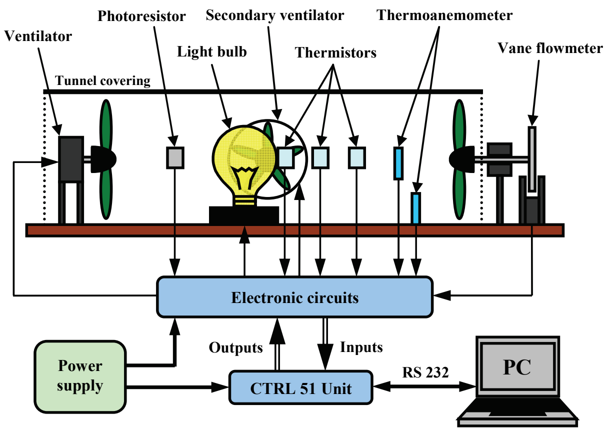


Fig. 2: Scheme of Hot-Air Tunnel and Whole Control System

The CTRL 51 unit has been produced by Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic [11] and it has the following technical parameters [10]:

- CPU Intel 8751
- 4 KB internal EPROM
- 128 B internal + 256 B external RAM
- 16 analog inputs and 4 analog outputs
- A/D converter with 0-10 V range and 12 bit resolution
- D/A converter with 0-10 V range, 12 bit resolution and no more than 3 % of mutual influence
- Communication with PC via standard serial interface RS232 (parameters: max. speed 9600 Bd, 8 data bits, 1 stop bit, without parity)
- Power voltage +5 V at current consumption 0.6 A and +15 V at 0.1 A
- Outer dimensions approximately 6 x 17 x 21 cm

The Table 1 and Table 2 denote the meaning of input and output channels of CTRL 51 unit, respectively.

Table 1: Connection of Input Signals of CTRL 51 Unit

| Input channel | Sensor |
|---|---|
| Input 1 ($y1$) | Light intensity of the bulb (photoresistor) |
| Input 2 ($y2$) | Temperature a few mm from the bulb (2nd thermistor) |
| Input 3 ($y3$) | Temperature of the bulb (1st thermistor) |
| Input 4 ($y4$) | Temperature at the end of the tunnel (3rd thermistor) |
| Input 6 ($y6$) | Airflow speed (thermoanemometer) |
| Input 7 ($y7$) | Airflow speed (vane flowmeter) |

Table 2: Connection of Output Signals of CTRL 51 Unit

| Output channel | Actuator |
|---|---|
| Output 1 ($u1$) | Bulb voltage (control of light intensity and bulb temperature) |
| Output 2 ($u2$) | Voltage of the primary ventilator (control of revolutions) |
| Output 3 ($u3$) | Voltage of the secondary ventilator (control of revolutions) |

The considered loop covers primary ventilator voltage $u2$ (control signal), which influences airflow speed measured by vane flowmeter $y7$ (controlled variable). The other actuating signals were preset to constant values – bulb voltage $u1$ to

0 V and secondary ventilator voltage $u3$ also to 0 V. Besides, analysis of another possible control loop $u1$-$y3$ using a robust approach can be found in [12] (control of bulb temperature $y3$ via voltage $u1$).

All presented identification and control experiments were performed using the notebook HP Compaq nc6120 with Intel Pentium M processor 1.86 GHz, 512 MB DDR-333 SDRAM, Windows XP and MATLAB 6.5.1. The communication between MATLAB and CTRL 51 unit was arranged through four user functions (for initialization, reading and writing the data, and for closing) and the synchronization of the program with real time was done via „semaphore“ principle (furthermore, the utilization of MATLAB functions „tic“ and „toc“ as an alternative were tested). To ensure the sufficient emulation of the continuous-time control algorithms, the sampling time 0.1 s was set. The detailed information about utilization of serial link under MATLAB including mentioned user routines, program synchronization mechanism and several tests can be found in [13]. The discretization of integrative part of control laws was carried out by left rectangle approximation method (the trapezoid method was also tried with the very similar results).

## 3 Model Identification

Naturally, the first task was to determine static and dynamic behaviour of the controlled system in order to have a suitable mathematical model for control synthesis.

The static characteristic is shown in Fig. 3. The depicted points have been obtained by measuring the steady outputs $y7$ for 10 values of input $u2$ (actually, the points are averages of last 20 measured “steady” values with sampling time 0.2 s).

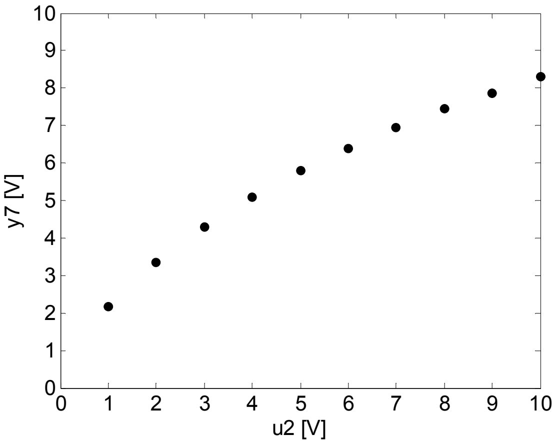


Fig. 3: Static characteristics

As can be seen, the system is nonlinear, because the static characteristic is not a straight line.

Next, the working point where the plant can be considered linear has been chosen as $y7 \in \langle 5\text{V}; 6\text{V} \rangle$ for all following control experiments. Thus the corresponding input signals have been determined to cover the intended area and subsequently the step response for change of $u2$ from 3V to 6V has been measured.

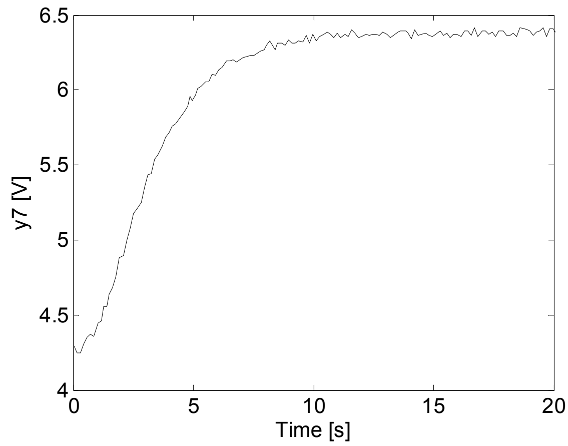


Fig. 4: Measured step response

The curve from Fig. 4 was normalized to unit input step change and approximated by response of second order system:

$$G(s) = \frac{K}{(Ts+1)^2} \qquad (1)$$

where $K$[-] is a gain and $T$[$s$] is a double time constant. The size of the gain can be obtained both from static characteristics and step response. The identification of time constant has been done using standard least squares method. So, the final transfer function of the controlled plant has been identified as:

$$G(s) = \frac{0.7026}{(1.7756s+1)^2} = \frac{0.2228}{s^2 + 1.1264s + 0.3172} \qquad (2)$$

The comparison of measured (normalized) and identified step responses indicating concordance of real plant with its mathematical model is depicted in Fig. 5.

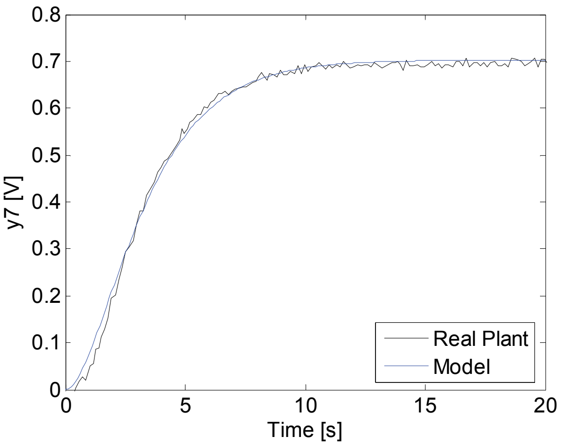

Fig. 5: Comparison of step responses

## 4 Control Design Methods: Simulations vs. Real Behaviour

A number of various PI/PID controller design methods have been applied to control of the airflow speed in hot-air tunnel. The range of analyzed techniques covers both classical methods from mid of the last century and recent synthesis methodologies. For each method, the controller has been designed, tested under simulative conditions in Matlab/Simulink environment and subsequently applied to the real plant.

The Fig. 6 shows basic control loop in Simulink. The block "Constant" is used to add some initial output value and approximately adjust the linearized model behaviour to the respective working point.

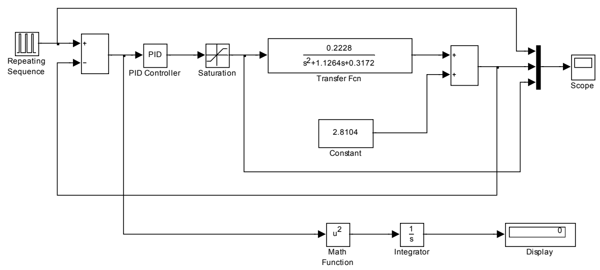

Fig. 6: Simulink scheme

Not all the used methods are able to work directly with the identified model in the form of (2). Due to this, the model has to be transformed also to the standard three-parameter transfer function:

$$G(s)=\frac{K}{T_1 s+1}e^{-Ls}=\frac{0.7026}{3.81s+1}e^{-0.5s} \quad (3)$$

The gain $K=0.7026[-]$, time constant $T_1=3.81[s]$ and fictive dead time $L=0.5[s]$ have been determined on the basis of step responses from Fig. 5 $\left(y(T_1)=0.6321K\right)$.

The original second order model (2) has been used by Whiteley standard forms, Naslin method and polynomial synthesis. On the other hand, Cohen-Coon method, Fruehauf method, Balanced tuning of PI controllers and desired model method have exploited first-order-plus-time-delay model (3). Furthermore, Chien-Hrones-Reswick method and step response method utilized also reading some data directly from the step response (delay time, rise time).

Nevertheless, the paper simply can not present the complete theoretical background and computation details for all employed methods. Thus, it usually contains only the main ideas with key references and then the transfer functions describing the final controllers immediately follow.

### 4.1 Chien-Hrones-Reswick Method

The first control design method is the traditional Chien-Hrones-Reswick tuning for PI and PID controllers [14]. The closed-loop response without any overshoot has been assumed. The resulting feedback PI controller is given by:

$$C_{PI}(s)=4.7987+\frac{1.0496}{s} \quad (4)$$

while the transfer function of the ideal PID controller can be computed as:

$$C_{PID}(s)=8.2263+\frac{2.1591}{s}+2.0566s \quad (5)$$

The simulative and real control results for PI controller are shown in Fig. 7 and Fig. 8, respectively. The Figs. 9 and 10 then present behaviour obtained using PID algorithm.

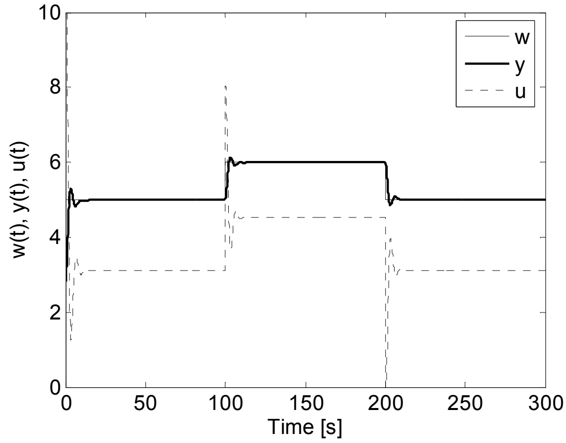

Fig. 7: Chien-Hrones-Reswick (PI, simulation)

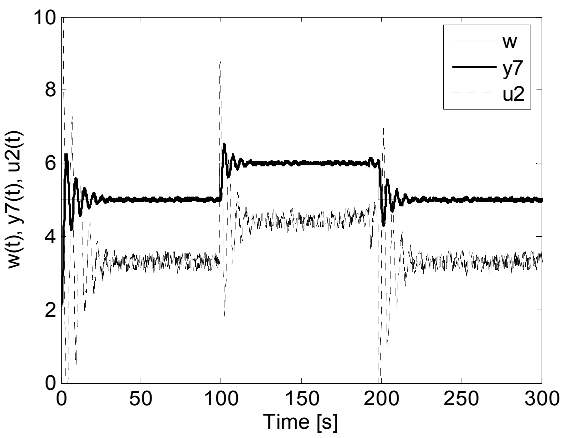

Fig. 8: Chien-Hrones-Reswick (PI, real)

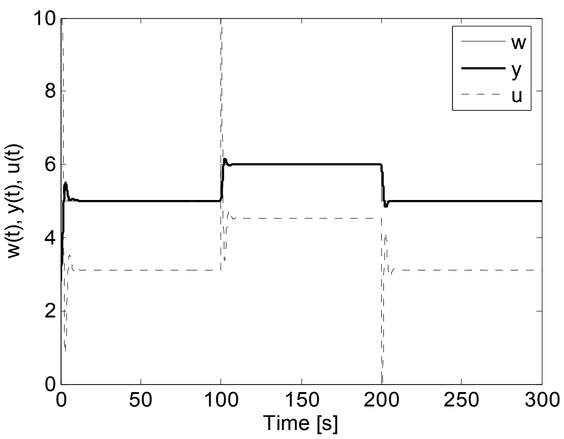

Fig. 9: Chien-Hrones-Reswick (PID, simulation)

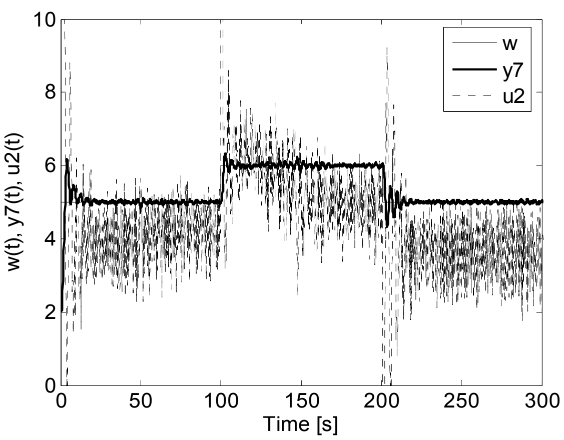

Fig. 10: Chien-Hrones-Reswick (PID, real)

## 4.2 Cohen-Coon Method

Another conventional approach to PI(D) controller calculation provides the Cohen-Coon method [15], known also as the method of 1/4 damping. The regulators tuned in this way are given by:

$$C_{PI}(s) = 9.8795 + \frac{7.5572}{s} \tag{6}$$

$$C_{PID}(s) = 14.8164 + \frac{12.6981}{s} + 2.6311s \tag{7}$$

As can be seen from Figs. 11 – 14, neither PI nor PID controllers were able to stabilize the real control loop, despite their "acceptable" simulation results.

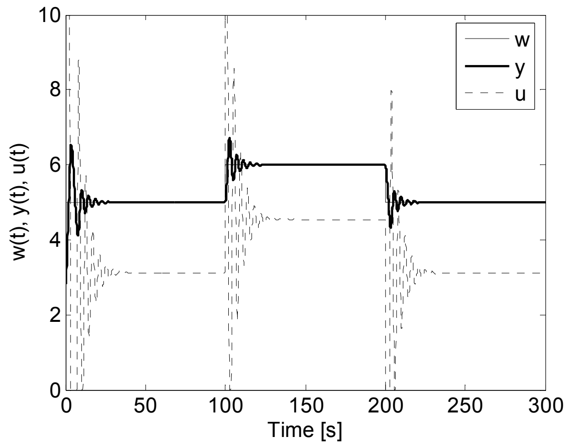

Fig. 11: Cohen-Coon (PI, simulation)

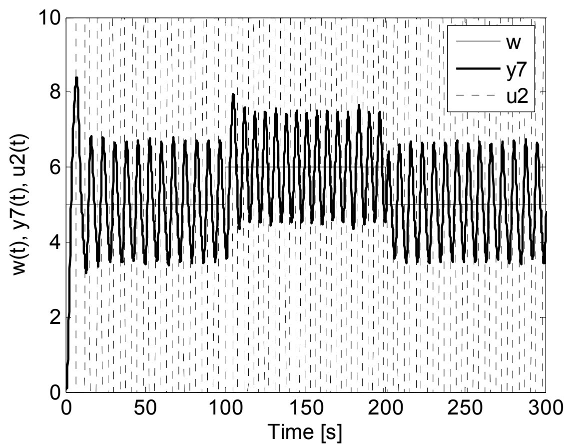

Fig. 12: Cohen-Coon (PI, real)

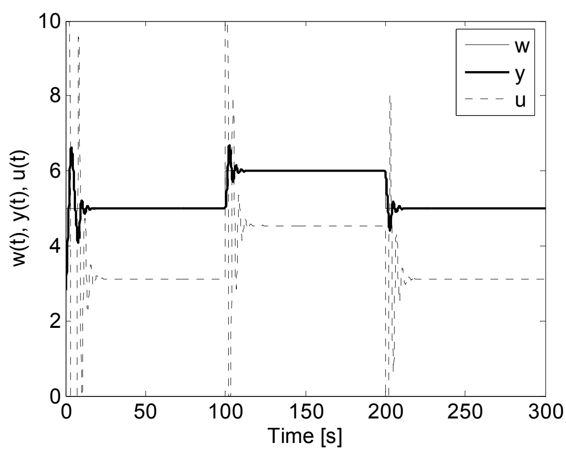

Fig. 13: Cohen-Coon (PID, simulation)

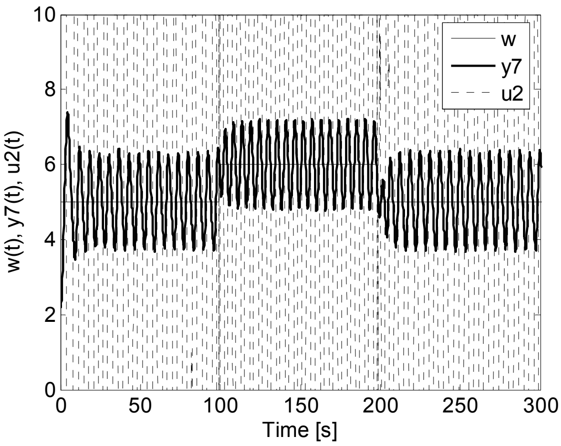


Fig. 14: Cohen-Coon (PID, real)

## 4.3 Step Response Method

There is an array of recommendations for direct step-response-based tuning of controllers [16]. One of them is applied also for the purpose of this paper, i.e. the compensators are:

$$C_{PI}(s) = 12.3395 + \frac{7.0511}{s} \tag{8}$$

$$C_{PID}(s) = 17.1381 + \frac{17.1381}{s} + 4.2845s \tag{9}$$

Unfortunately, as in the previous Cohen-Coon method, the designed controllers have not brought suitable real outputs (see Figs. 15 – 18).

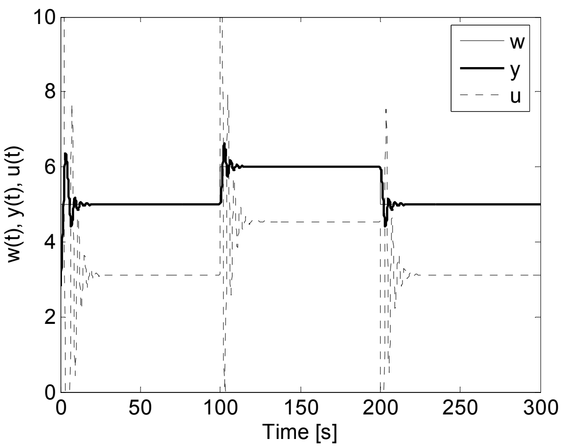


Fig. 15: Step response method (PI, simulation)

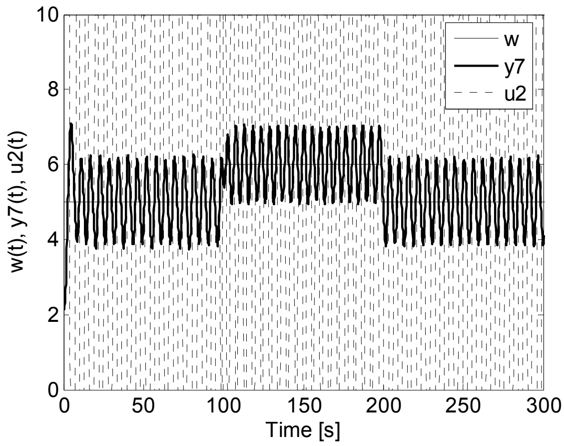


Fig. 16: Step response method (PI, real)

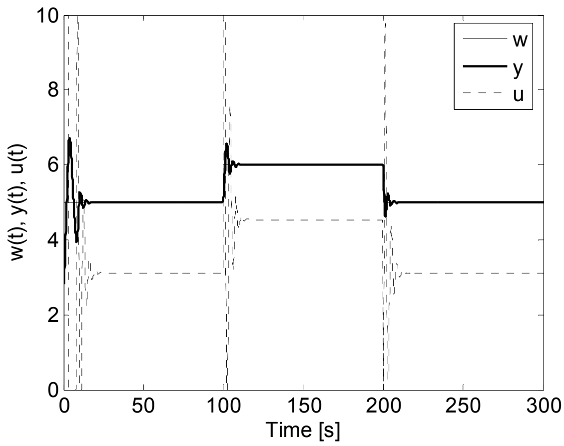


Fig. 17: Step response method (PID, simulation)

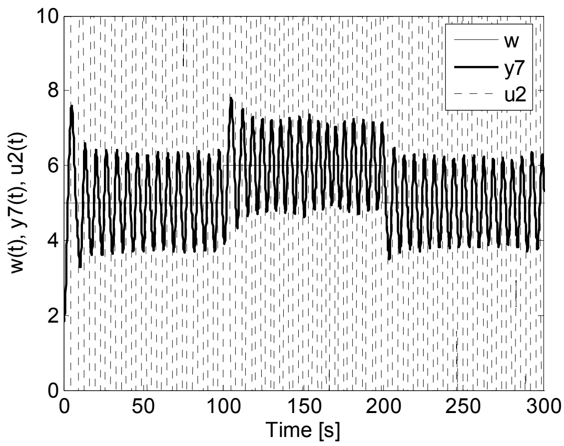


Fig. 18: Step response method (PID, real)

## 4.4 Whiteley Standard Forms Method

Main idea of all methods using standard forms is to adjust the transfer function of the closed loop into some prescribed form which is known to provide appropriate control results. There are more kinds of such paradigmatic closed-loop transfer functions in literature [2], [17]. One of them are so-called Whiteley standard forms (examples of the other ones can be Kessler, Butterworth, Chebyshev, etc.). Nevertheless, it has led to the PI controller with negative proportional part. That is why only PID controller:

$$C_{PID}(s) = 4 + \frac{0.3438}{s} + 7.7154s \tag{10}$$

has been used for experiments from Fig. 19 and Fig. 20.

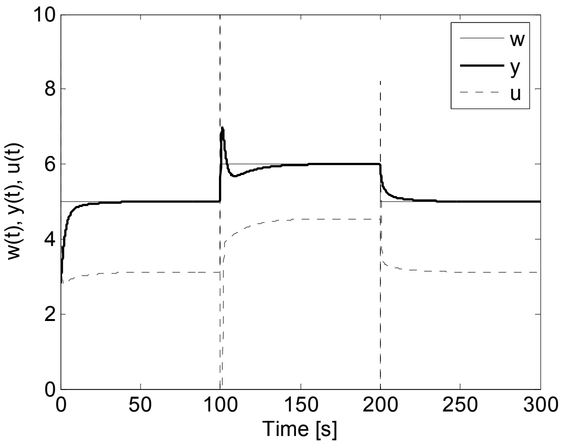


Fig. 19: Whiteley standard forms (PID, simulation)

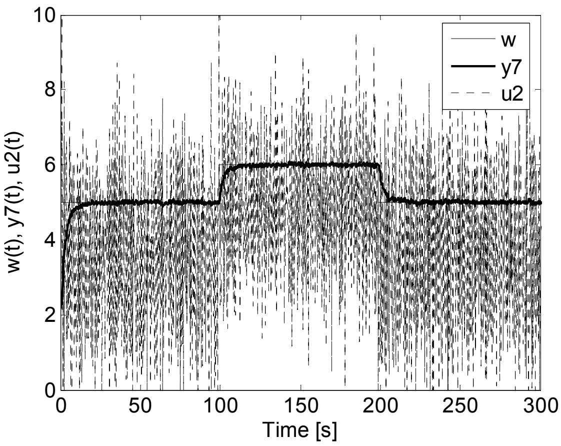


Fig. 20: Whiteley standard forms (PID, real)

## 4.5 Naslin Method

Next studied control design method was invented by Naslin [17]. Maximum 5% overshoot has been considered for both PI and PID controller synthesis:

$$C_{PI}(s) = 1.4232 + \frac{0.8016}{s} \tag{11}$$

$$C_{PID}(s) = 1.5 + \frac{0.8342}{s} + 0.0673s \tag{12}$$

Figs. 21 – 24 show corresponding control results.

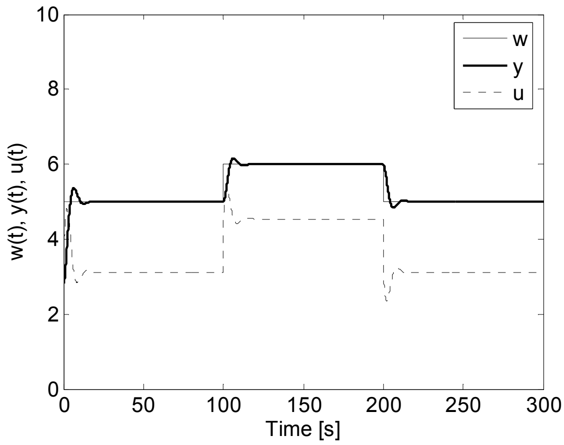


Fig. 21: Naslin (PI, simulation)

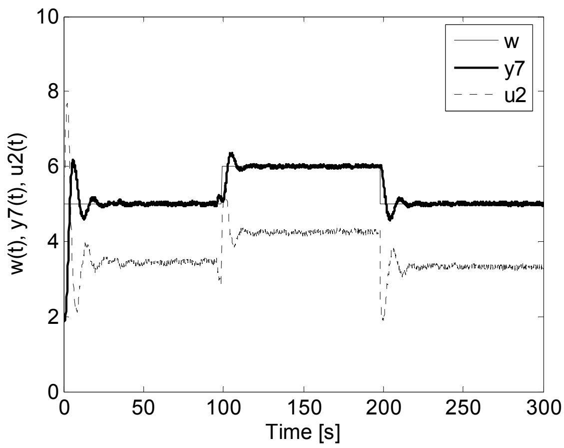


Fig. 22: Naslin (PI, real)

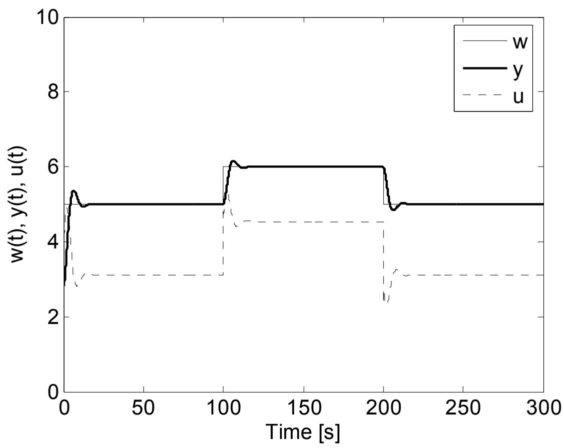


Fig. 23: Naslin (PID, simulation)

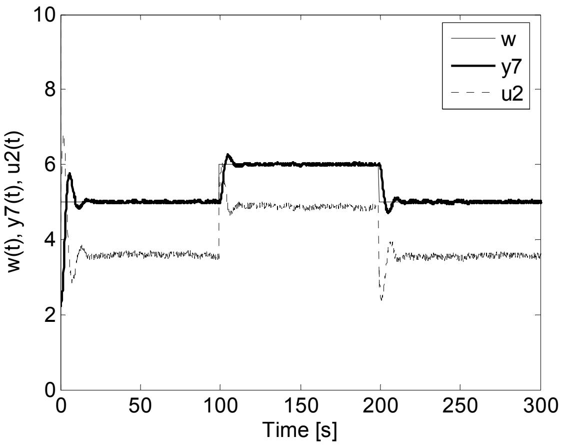

Fig. 24: Naslin (PID, real)

## 4.6 Fruehauf Method

Further, Fruehauf method ([18], [19]) has been used for PI:

$$C_{PI}(s) = 6.0252 + \frac{2.4101}{s} \qquad (13)$$

and PID:

$$C_{PID}(s) = 6.0252 + \frac{2.4101}{s} + 1.5063s \qquad (14)$$

control design. The obtained curves are shown in Figs. 25 – 28. The interesting output is depicted in Fig. 26, where real closed-loop stabilization depends on the specific working point.

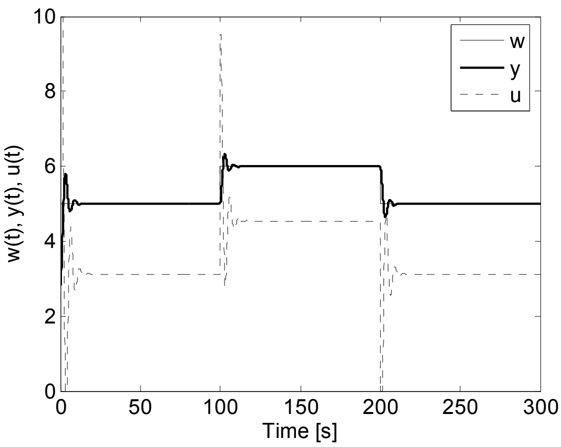

Fig. 25: Fruehauf (PI, simulation)

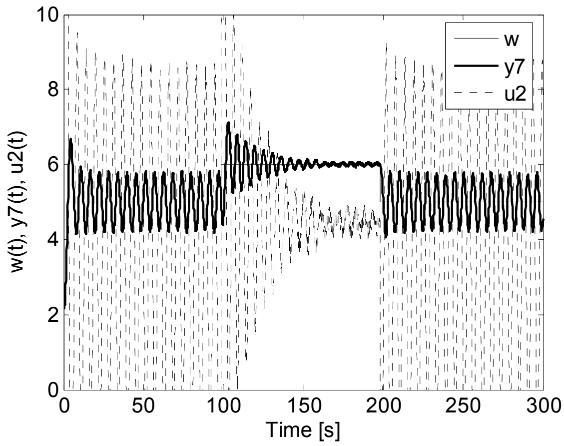

Fig. 26: Fruehauf (PI, real)

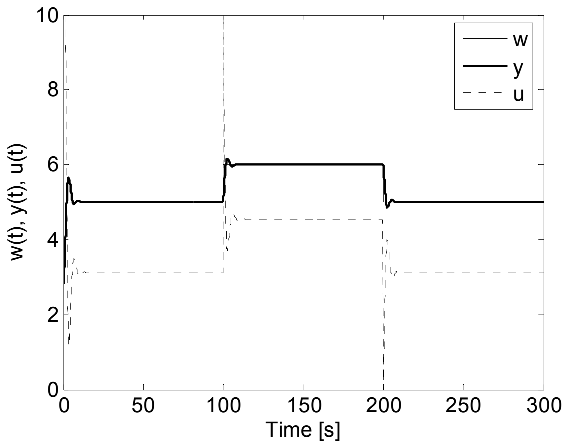

Fig. 27: Fruehauf (PID, simulation)

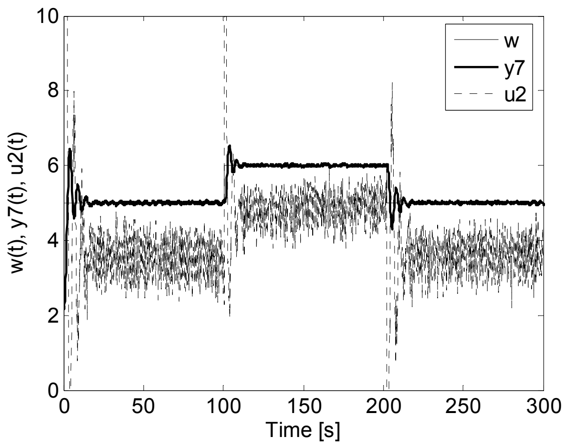

Fig. 28: Fruehauf (PID, real)

## 4.7 Balanced Tuning of PI Controllers

One of relatively recent methods is balanced tuning of PI controllers which preserves actuators [20], [21]. The designed regulator is:

$$C_{PI}(s) = 1.2593 + \frac{0.328}{s} \qquad (15)$$

and control results are depicted in Fig. 29 and Fig. 30.

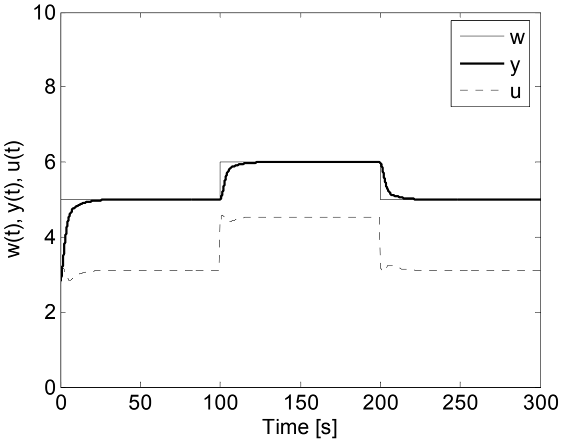


Fig. 29: Balanced tuning (PI, simulation)

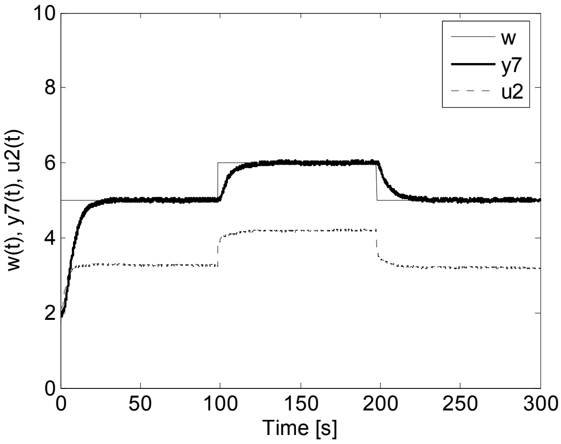


Fig. 30: Balanced tuning (PI, real)

### 4.8 Desired Model Method

Another comparatively new tuning technique for both continuous-time and discrete-time controllers is represented by desired model method (formerly known as inversion dynamics method) [22] which gives:

$$C_{PI}(s) = 1.0473 + \frac{0.2749}{s} \qquad (16)$$

$$C_{PID}(s) = 6.7237 + \frac{1.0473}{s} + 9.4777s \qquad (17)$$

The Figs. 31 – 34 present control outputs.

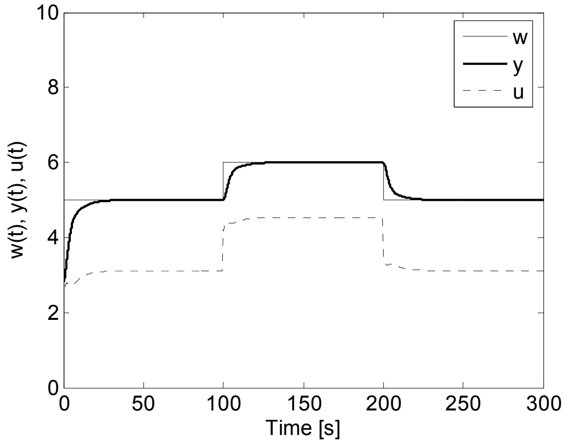


Fig. 31: Desired model (PI, simulation)

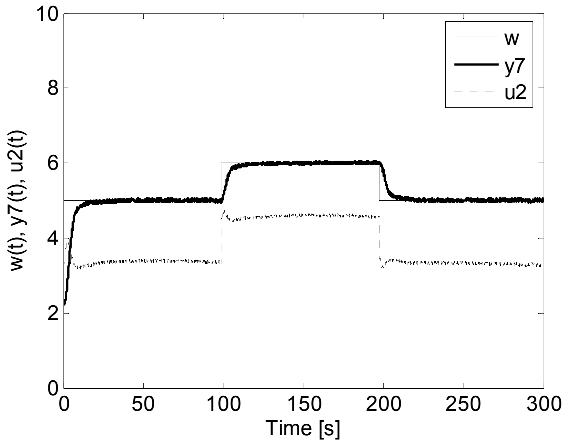


Fig. 32: Desired model (PI, real)

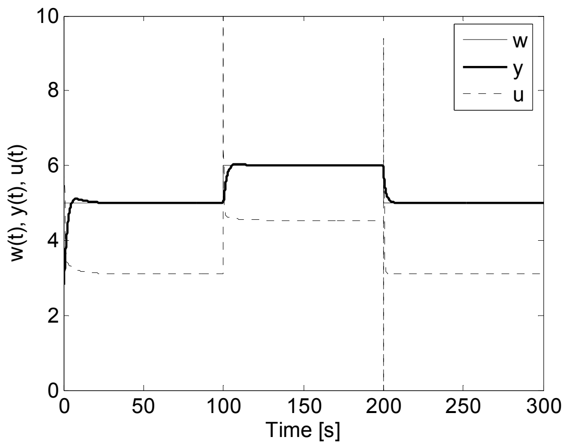


Fig. 33: Desired model (PID, simulation)

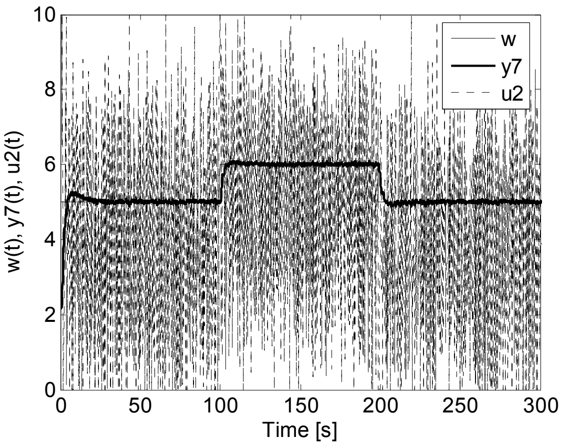


Fig. 34: Desired model (PID, real)

### 4.9 Polynomial Synthesis

Last but not least, a polynomial approach to control design has been employed [23]. The second order transfer function of the controlled plant, one-degree-of-freedom (1DOF) control loop configuration, stepwise reference signal and multiple root of closed-loop characteristic polynomial $m = 0.5$ have led to realistic PID controller:

$$C_{PID}(s) = \frac{0.8922s^2 + 1.0004s + 0.2805}{s(s + 0.8736)} \quad (18)$$

The control results are shown in Fig. 35 and Fig. 36.

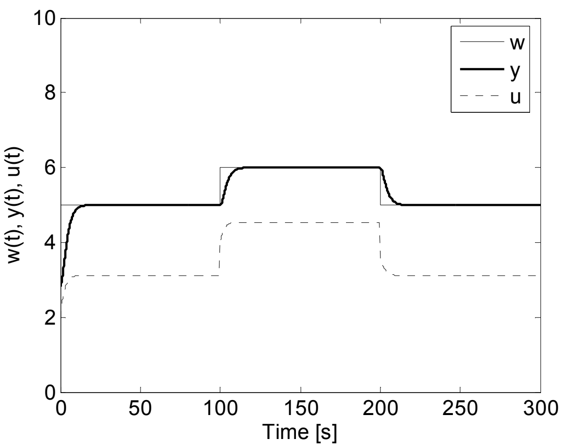


Fig. 35: Polynomial synthesis (PID, simulation)

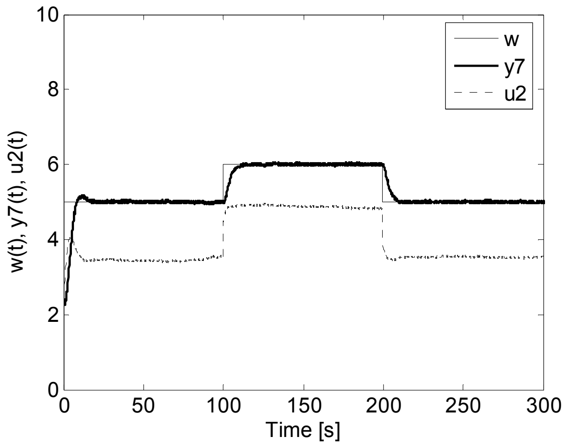


Fig. 36: Polynomial synthesis (PID, real)

## 5 Conclusion

The comparison of all studied control design methods has been accomplished using Integrated Squared Error (ISE) criterion:

$$\mathrm{ISE} = \int_0^\infty e^2(t)dt \quad (19)$$

and by means of maximum overshoots of the controlled variable. The results can be seen in Table 3 and Table 4.

Table 3: Comparison of ISE calculations

| Control design method | Simul. | Real |
|---|---|---|
| Chien-Hrones-Reswick – PI | 6.3878 | 19.1507 |
| Chien-Hrones-Reswick – PID | 5.6986 | 19.6456 |
| Cohen-Coon – PI | 16.3335 | x |
| Cohen-Coon – PID | 12.0965 | x |
| Step response – PI | 10.7072 | x |
| Step response – PID | 13.0737 | x |
| Whiteley – PID | 11.6172 | 19.0597 |
| Naslin – PI | 11.9440 | 26.8046 |
| Naslin – PID | 11.6100 | 19.5700 |
| Fruehauf – PI | 7.3606 | 77.5342 |
| Fruehauf – PID | 6.5187 | 17.7442 |
| Balanced tuning – PI | 15.7500 | 55.5557 |
| Desired model – PI | 18.3333 | 27.0085 |
| Desired model – PID | 6.7306 | 15.6877 |
| Polynomial synthesis – PID | 18.5853 | 29.5254 |

Table 4: Comparison of maximum overshoots

| Control design method | Simul. | Real |
|---|---|---|
| Chien-Hrones-Reswick – PI | 0.2945 | 1.2466 |
| Chien-Hrones-Reswick – PID | 0.4917 | 1.1905 |
| Cohen-Coon – PI | 1.6933 | x |
| Cohen-Coon – PID | 1.6165 | x |
| Step response – PI | 1.3630 | x |
| Step response – PID | 1.7229 | x |
| Whiteley – PID | 0 | 0.0806 |
| Naslin – PI | 0.3569 | 1.1832 |
| Naslin – PID | 0.3678 | 0.7607 |
| Fruehauf – PI | 0.8073 | 1.6838 |
| Fruehauf – PID | 0.6364 | 1.4518 |
| Balanced tuning – PI | 0 | 0.0696 |
| Desired model – PI | 0 | 0.0586 |
| Desired model – PID | 0.1082 | 0.2772 |
| Polynomial synthesis – PID | 0 | 0.1893 |

In simulations, the Chien-Hrones-Reswick, Fruehauf and desired model method (PID version) belong among the ones with the lowest values of ISE. Under the real conditions, also Naslin method and Whiteley standard forms joined those three with good results. Cohen-Coon and step response methods were not able to stabilize the closed loop at all. On the whole, the PID controller designed via desired model method is practically the best from the ISE viewpoint.

The confrontation of maximum overshoots brought the simulation victory for balanced tuning of PI controllers, Whiteley standard forms, desired model method (PI) and polynomial synthesis. Actually, the PI from desired model method achieved the lowest overshoot also during control of real plant. Further, it has been followed by balanced tuning and Whiteley standard forms (however very aggressive control signal has been the drawback of the Whiteley).

*Acknowledgements:*
The work was supported by the Ministry of Education, Youth and Sports of the Czech Republic under Research Plan No. MSM 7088352102. This assistance is very gratefully acknowledged. Moreover, the authors would like to thank Jana Vyoralová for providing the data measured under the scope of her master's thesis [24].

*References:*
[1] J. G. Ziegler, N. B. Nichols, Optimum settings for automatic controllers, *Transactions of ASME*, Vol. 64, 1942, pp. 759-768.
[2] K. J. Åström, T. Hägglund, *PID Controllers: Theory, Design and Tuning*, Instrument Society of America, USA, 1995.
[3] A. O'Dwyer, *Handbook of PI and PID Controller Tuning Rules*, Imperial College Press, London, UK, 2003.
[4] S. Skogestad, Simple analytic rules for model reduction and PID controller tuning, *Journal of Process Control*, Vol. 13, No. 4, 2003, pp. 291-309.
[5] K. J. Åström, T. Hägglund, The future of PID control, *Control Engineering Practice*, Vol. 9, No. 11, 2001, pp. 1163–1175.
[6] T. Emami, J. M. Watkins, A Unified Approach for Sensitivity Design of PID Controllers in the Frequency Domain, *WSEAS Transactions on Systems and Control*, Vol. 4, No. 5, 2009, pp. 221-231.
[7] T. Emami, J. M. Watkins, Robust Performance Characterization of PID Controllers in the Frequency Domain, *WSEAS Transactions on Systems and Control*, Vol. 4, No. 5, 2009, pp. 232-242.
[8] B. Šulc, S. Vrána, Some Observations on Development and Testing of a Simple Autotuning Algorithm for PID Controllers, *WSEAS Transactions on Systems and Control*, Vol. 4. No. 10, 2009, pp. 497-508.
[9] S.-J. Huang, Y.-H. Lo, Metal Chamber Temperature Control by Using Fuzzy PID Gain Auto-tuning Strategy, *WSEAS Transactions on Systems and Control*, Vol. 4, No. 1, 2009, pp. 1-10.
[10] L. Smutný, J. Škuta, J. Farník, Model of hot-air circuit (Model teplovzdušného obvodu), Technical report to HS 311107 "Technická pomoc při návrhu a zhotovení modelu teplovzdušného obvodu", VSB – Technical University of Ostrava, Czech Republic, 2002. (In Czech).
[11] P. Klán, D. Honc, J. Jindřich, New measuring unit CTRL V3 (Nová měřicí jednotka CTRL V3), In: *Proceedings of conference MATLAB 2003*, Prague, Czech Republic, 2003. (In Czech).

[12] R. Matušů, R. Prokop, M. Dlapa, Robust Control of Temperature in Hot-Air Tunnel, In: *Proceedings of the 16th Mediterranean Conference on Control and Automation*, Ajaccio, France, 2008, pp. 576-581, ISBN 978-1-4244-2505-1.
[13] F. Dušek, D. Honc, The use of serial link under MATLAB 6 (Využití sériové linky pod MATLABem verze 6), In: *Proceedings of conference MATLAB 2002*, Prague, Czech Republic, 2002. (In Czech).
[14] K. L. Chien, J. A. Hrones, J. B. Reswick, On the automatic control of generalized passive systems, *Transactions of ASME*, Vol. 74, 1952, pp. 175-185.
[15] K. H. Cohen, G. A. Coon, Theoretical consideration of retarded control, *Transactions of ASME*, Vol. 75, 1953, pp. 827-834.
[16] K. J. Åström, T. Hägglund, Revisiting the Ziegler–Nichols step response method for PID control, *Journal of Process Control*, Vol. 14, No. 6, 2004, pp. 635-650.
[17] J. Balátě, Automatic control (Automatické řízení), BEN – technická literatura, Prague, Czech Republic, 2003. (In Czech).
[18] A. O'Dwyer, A Summary of PI and PID Controller Tuning Rules for Processes with Time Delay. Part I. In: *Proceedings of the IFAC Workshop on Digital Control: Past, Present and Future of PID Control*, Terrassa, Spain, 2000, pp. 175-180.
[19] A. O'Dwyer, A Summary of PI and PID Controller Tuning Rules for Processes with Time Delay. Part II. In: *Proceedings of the IFAC Workshop on Digital Control: Past, Present and Future of PID Control*, Terrassa, Spain, 2000, pp. 242-247.
[20] P. Klán, R. Gorez, Balanced tuning of PI controllers, *European Journal of Control*, Vol. 6, No. 6, 2000, pp. 541-550.
[21] P. Klán, Tuning of PI controllers preserving actuators (Nastavení PI regulátorů chránící akční členy), *Automa*, Vol. 11, No. 2, 2005, pp. 50-52. (In Czech).
[22] M. Vítečková, *Tuning of controllers by dynamics inversion method* (*Seřízení regulátorů metodou inverze dynamiky*), VSB – Technical University of Ostrava, Czech Republic, 2000. (In Czech).
[23] V. Kučera, Diophantine equations in control – A survey, *Automatica*, Vol. 29, No. 6, 1993, pp. 1361-75.
[24] J. Vyoralová, *Experimental verification of conventional controller design methods during regulation of laboratory model of air-heating tunnel* (*Experimentální ověření metod návrhu konvenčních regulátorů při řízení laboratorního modelu teplovzdušného tunelu*), Master's Thesis, Tomas Bata University in Zlín, Czech Republic, 2009. (In Czech).