H_∞ Multivariable PID Control of a Thermal Plant

Bozhidar Rakov
Faculty of Automatics
Technical University of Sofia
Sofia, Bulgaria
brakov@tu-sofia.bg

Georgi Ruzhekov
Faculty of Automatics
Technical University of Sofia
Sofia, Bulgaria
g_ruzhekov@tu-sofia.bg

***Abstract* — The current paper investigates the possibility to design a discrete multivariable H_∞ PID controller using the MATLAB *hinfstruct()* function. Two PIDs are designed. They are compared against a full order H_∞ controller and a decentralized PID with a static decoupling matrix. Frequency response of the singular values of the closed loop sensitivity functions are calculated and compared. All controllers are tested on a laboratory model, representing a thermal plant. The control algorithms are implemented on a Programmable Logic Controller. The elements of the multivariable PIDs are implemented using velocity form for simpler anti wind-up mechanism realization. The parameters are calculated by MATLAB and then transferred using OPC connection to a SCADA system. Two sets of experiments are conducted with the real system, testing two different sets of reference signal changes. Based on known performance criteria, a comparison is made. The results show the viability of a fixed order, fixed structure controller designed based on H_∞ theory.**

***Keywords* — H_∞, fixed structure, multivariable, PID, PLC**

I. INTRODUCTION

Most industrial installations today still depend on the PID control law for the lower level [1]. On one hand it is due to the simplicity of the controller. It consists mainly of three parameters. These parameters have physical meaning and are easily understood by engineers and maintenance personnel of the operating plants. On the other hand, it is due to all the accumulated knowledge on the topic and all the existing hardware and software solutions. Despite that, the application of the PID control law into the multivariable case is not simple. For a long time, there was no general solution or framework, describing a procedure or algorithm for parameter calculation.

Parallel to the PID control theory, as early as the 80s the H_∞ theory has been developing rapidly [2]. It attracts many researchers. A general solution was found through numerical calculations. Controllers designed using the H_∞ theory might be optimal or suboptimal in the sense of norm H_∞. One disadvantage of the general H_∞ theory framework was that the received controllers cannot have fixed structure *a priori*. That led to research in this direction and some solutions in this regard were proposed. Now it is even possible to design a PID controller based on a H_∞ criterion.

The present article would like to try and combine the PID controller and the H_∞ theory in application to the multivariable case for a plant having simple dynamics.

II. H_∞ CONTROLLER DESIGN

The plant to be controlled is shown of Fig. 1. It is a thermal plant consisting of 4 pieces of aluminum.

Fig. 1. Thermal plant

At each piece a thermocouple is attached. On three of them a heater is used for control and on the fourth a fan. An identification procedure based on the subspace method [3] is conducted and a discrete multivariable state space plant with the three heaters acting as inputs is obtained. The outputs to be controlled are the temperatures of three of the aluminum pieces. The model has nine states. Step response with 3 standard deviations confidence interval and frequency response of the singular values of the identified plant are shown on Fig. 2 and Fig. 3

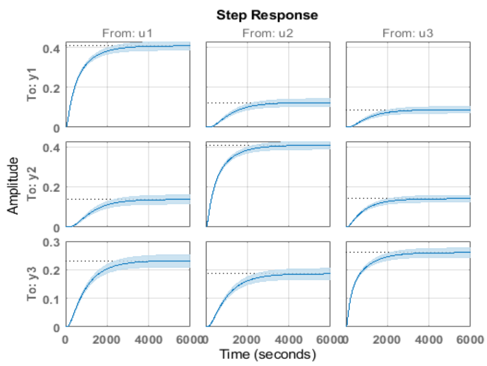

Fig. 2. Step response of the plant

The plant is controlled in a standard negative feedback loop configuration as shown on Fig. 4.

The signal r is the reference, e is the control error, u is the control signal, d represents low frequency output disturbance, n is high frequency noise, y is the output of the plant G to be controlled by K. Basic equations describing the relation between the different signals are shown in (1) and (2)

$$y = T_o(r - n) + S_o d \quad (1)$$

$$u = KS_o(r - n - d) \quad (2)$$

T_o denotes the output complementary sensitivity function, S_o denotes the output sensitivity function and KS_o denotes the sensitivity at the plant input. T_o and S_o are formed by (3)

$$T_o = \frac{GK}{I+GK}, S_o = (I + GK)^{-1} \qquad (3)$$

Where I is the identity matrix. The current work uses mixed S/KS sensitivity scheme [4] for the design of H_∞ controller shown on Fig. 5.

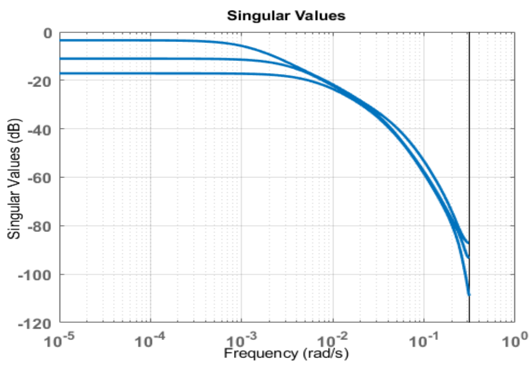

Fig. 3. Frequency response of the singular values of the plant

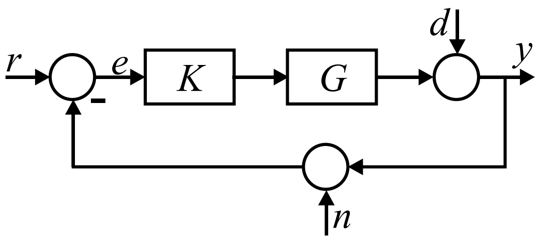

Fig. 4. Block-diagram of the closed loop system

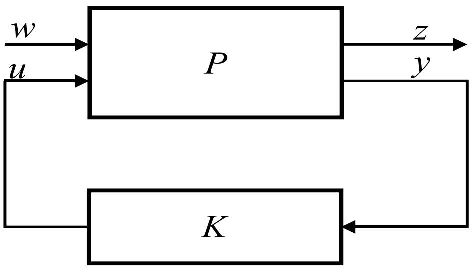

Fig. 5. Mixed *S/KS* design

The signal w represents the external input to the loop, z are the error signals or the signals acting as quality measures, u is the control signal, y is the feedback signal, K is the controller to be designed and P represents the extended plant. The extended plant consists of the plant to be controlled G and the transfer matrices of the weighting filters W_p and W_u. In this case the disturbance at the plant output d acts as external signal w. The plant output y in the presence of output disturbance and the control signal u act as error signals to be minimized. The input to the controller is the control error, due to existence of disturbance at the plant output. Then the problem is to find a stabilizing controller K such that (4):

$$\min_K \max_\omega \bar{\sigma}\big(F_l(j\omega)\big), \qquad (4)$$

where F_l is (5):

$$F_l = \begin{bmatrix} W_p S_o \\ W_u K S_o \end{bmatrix}. \qquad (5)$$

Furthermore, the desired controller K has a fixed order and a fixed structure as shown on (6):

$$K(z) = \begin{pmatrix} K_{11} & \cdots & K_{1m} \\ \vdots & \ddots & \vdots \\ K_{r1} & \cdots & K_{rm} \end{pmatrix}, \qquad (6)$$
$$K_{ij}(z) = \left(K_p + K_i \frac{T_s z}{z-1} + K_d \frac{1}{T_f + \frac{T_s z}{z-1}}\right)$$

Where K_{ij} is a discrete PID controller, having K_p as proportional gain, K_i as integral gain, K_d as derivative gain and T_f as filter constant. The solution for fixed order and fixed structure H_∞ controller could be found in [5]. The authors of [5] propose a method, based on no smooth techniques suited for H_∞ synthesis and for semi-infinite eigenvalue or singular value optimization programs. MATLAB introduced a function *hinfstruct()*, which implements the proposed techniques by Apkarian and Noll since version 2010b. In general the *hinfstruct()* function takes a fixed structure controller with tunable parameters and an extended plant P as outlined on Fig. 4. The output of the function is a tuned controller, minimizing the H_∞ norm of the closed loop structure, describing the connection between the external signals w and the error signals z. For further documentation the reader is pointed to [6].

Setting the weighting filters W_p and W_u as diagonal matrices with elements (7) and (8)

$$W_p(s) = \frac{\frac{s}{1.4} + 0.01}{s}, \qquad (7)$$

$$W_u(s) = 0.01\left(\frac{\frac{1}{0.1}s+1}{\frac{1}{10}s+1}\right) \qquad (8)$$

one receives a multivariable discrete PID controller. The filters are chosen in a way to provide a certain shape to the output sensitivity function S_o and KS_o. It is desired that the bandwidth of the closed loop system is no more than 0.01 rad/s and the maximum of the output sensitivity function not exceed 1.4. For the KS_o it is desired that the control amplitudes for signals with spectral components at around 0.1 rad/s not exceed 40dB. After that the limit decreases by 20 dB/decade until 10 rad/s to eliminate high control amplitudes for high frequency signals. Preliminary simulations showed unacceptable transient response of the closed loop system, when the reference change leads to control signal saturation. For this reason, a second PID controller is designed, with the goal of reduced control amplitudes at the expense of slower transients. The weighting filters for the second controller are diagonal matrices with elements (9) and (10):

$$W_p(s) = \frac{\frac{s}{1.4} + 0.006}{s}, \qquad (9)$$

$$W_u(s) = 0.06\left(\frac{\frac{1}{0.1}s+1}{\frac{1}{10}s+1}\right). \qquad (10)$$

The desired bandwidth of the closed loop system is decreased to 0.006 rad/s and the limit for low frequency signals is decreased to 24.43dB. Comparison of the frequency response of the singular values of the sensitivity functions of the closed loop systems for the two PID controllers are shown on Fig. 6 to Fig. 8.

For the output complementary sensitivity function T_o one could note that the first PID controller (annotated as *f* for “fast”) provides the closed loop system with a wider bandwidth. But this could lead to higher noise amplification at the system output, compared to the second controller (annotated as *s* for “slow”). Both systems lack high values for the maximum singular value of T_o, meaning that both systems should be robust to multiplicative model uncertainty.

The output sensitivity function S_o has values between 2.5 and 3 dB for the highest point of the maximum singular value for both systems. This values are well below the recommended limit of 6dB, leading to good stability margin. The feedback loop for the fast system is effective at around 0.004 rad/s, and for the slow one at around 0.0023 rad/s.

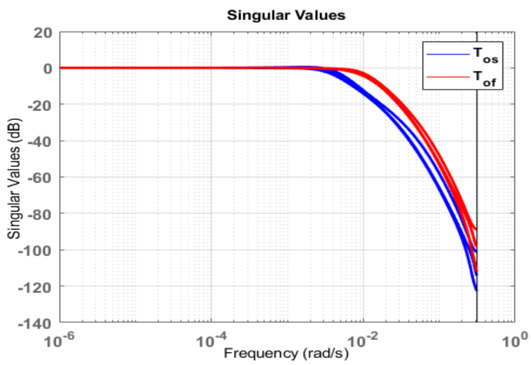

Fig. 6. Output complementary functions comparison

For the sensitivity at the plant input KS_o a difference in the high frequency region is observed. A high frequency noise would be amplified even further by the fast system. Additionally, a higher amplitude control signals are expected.

A good test for the fixed order controllers would be a comparison with a full order H_∞ controller and a PID controller with a decoupling matrix designed as outlined in [7].

For that reason, additional full order H_∞ controller is designed as well as a PID controller and a static decoupling matrix. Comparison of the frequency response of the singular values of the sensitivity functions of the closed loop systems for the fast PID and the full order H_∞ controller is shown on Fig. 9 to Fig. 11.

The singular values of the PID system are noted with a *p* subscript. The subscript *h* denotes the singular values of the H_∞ controller system. Excluding the higher values of the KS_o sensitivity of the PID controller, there is no big difference between the two systems.

Although the plant to be controlled has a simple, slow dynamics, there is one restrictive constraint, imposed by the control signals. The singular values of the scaled plant are shown on Fig. 12.

Based on [8] for perfect control the following requirement needs to be fulfilled (11):

$$\underline{\sigma}(G) > 1, \quad (11)$$

where $\underline{\sigma}(G)$ is the minimal singular value of the plant. Clearly this requirement is violated, meaning that for a sufficiently big reference signal the closed loop system would operate in the saturation limit, leading to deterioration of the overall performance.

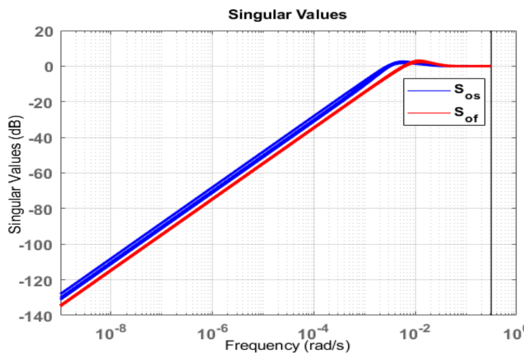

Fig. 7. Output sensitivity functions comparison

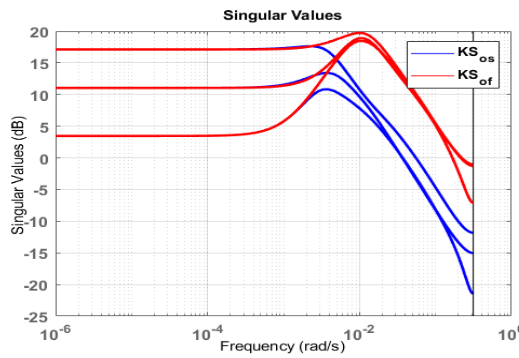

Fig. 8. Sensitivity at the plant input comparison

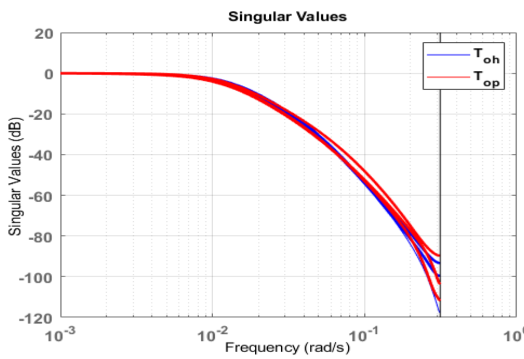

Fig. 9. Output complementary functions comparison

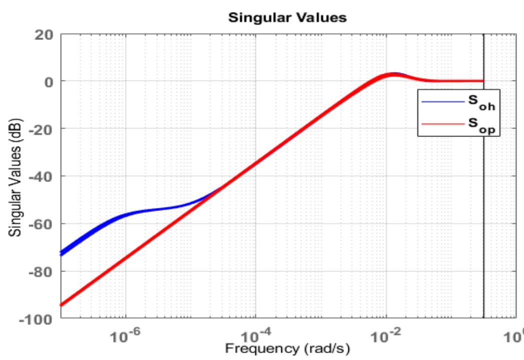

Fig. 10. Output sensitivity functions comparison

III. IMPLEMENTATION

The control algorithms and all supporting functions are developed with TIA Portal V18 and are executed by a Programmable Logic Controller (PLC). The elements of the multivariable PID controllers are implemented using a velocity form [8] as shown in (12).

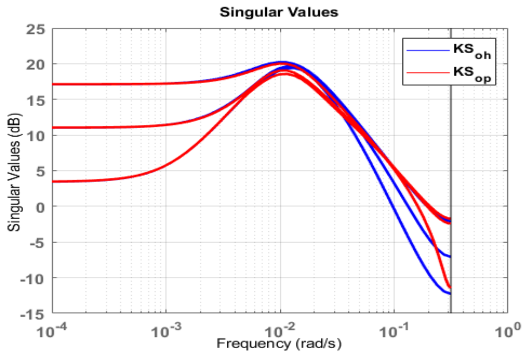

Fig. 11. Sensitivity at the plant input comparison

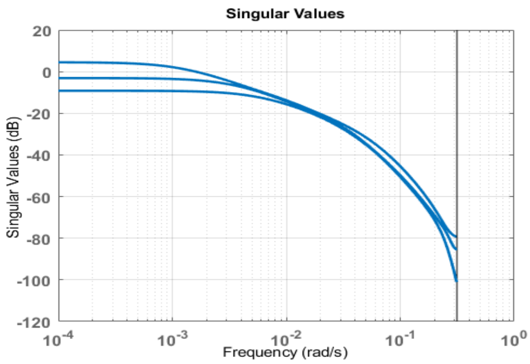

Fig. 12. Singular values of the scaled plant

$$K_{ij}(z) = \frac{T_s z}{z-1}\left(K_p \frac{z-1}{T_s z} + K_i + K_d \frac{z-1}{T_s z\left(T_f + \frac{T_s z}{z-1}\right)}\right). \qquad (12)$$

The use of (12) is dictated by the need for a simple anti wind-up solution. If one factorizes the discrete integral term for every control signal as shown on Fig. 13 then the solution to the anti-wind-up problem in this case would be to simply stop integrating in the presence of saturation.

The integral and derivative parts are implemented by backward Euler difference approximation. The sample time is set to ten seconds. As outlined earlier the design procedure of the controllers is carried out by MATLAB using *hinfstruct()* function. The values for the proportional, integral and derivative terms are send to the PLC through OPC connection.

A SCADA system is developed using WinCC Unified V18. It provides process visualization, possibility for mode of operation change (auto/manual), collecting signals values in a text file for further analysis, initiating controller parameters transfer etc.

IV. EXPERIMENTAL RESULTS

All controllers are tested in a closed loop system experiments, in regard to reference change. Two cases are tried. The first one leads to different amplitudes of the reference for all loops. The change of the reference signals varies between 30 and 40 degrees. Results are shown on Fig. 14 to Fig. 19. The notation is the following, *DPID* is the PID controller and the decoupling matrix, H_{inf} is the H_∞ controller, $MPID_f$ is the "fast" multivariable PID and $MPID_s$ is the "slow" one, *ref* represents the reference signal. Some variance in the initial conditions might be observed due to inability to control the environment and the temperature in the room.

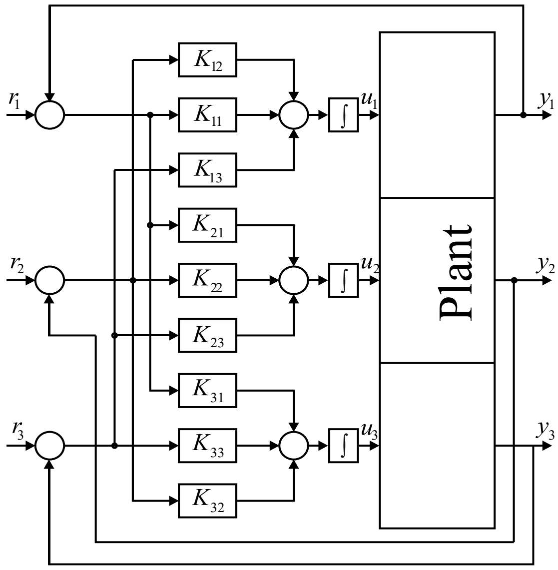

Fig. 13. Control structure of the closed loop system

All systems, excluding the "fast" PID have comparable performance if we neglect the difference in overshoot. The outlier mentioned exhibits strange behavior.

The control signal of the "fast" PID quickly exits the saturation limit and continues to decrease despite the existence of a big control error. This phenomenon leads to slower transient response and smaller control signal compared to the rest. One should note that this result is the opposite of what the frequency response of the singular values of the different sensitivity functions showed earlier. That is a good example of the dangers operating a closed loop system into the saturation limit.

Some performance criteria are calculated and shown in TABLE I, where overshoot, S_e and S_u are defined as (13), (14) and (15).

$$\sigma = \frac{(y(max) - y(\infty))}{(y(\infty) - y(0))} 100 \qquad (13)$$

$$S_e = \sqrt{\frac{1}{T}\sum_0^T e^2(k)}, k = 0, \ldots, (N-1), \qquad (14)$$

$$S_u = \sqrt{\frac{1}{T}\sum_0^T u^2(k)}, k = 0, \ldots, (N-1), \qquad (15)$$

where T is the time frame of the experiment, e is the control error and u is the control signal, k is the discrete time step and N is the number of samples. The rise time, settling time and overshoot are calculated using *stepinfo()* MATLAB function.

Looking at the rise time, the H_∞ controller and the decoupling PID are close, but if one accounts for the settling time the first one is clearly better in terms of fast response. On top of that the overshoot for the H_∞ controller is quite lower at around 2 to 3 %.

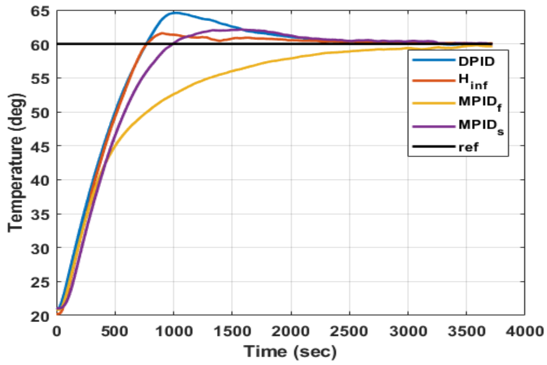

Fig. 14. Output 1 – comparison

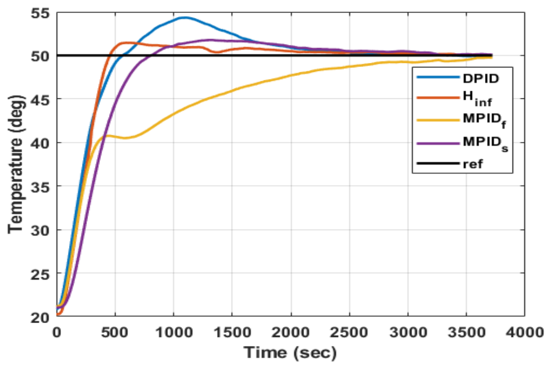

Fig. 15. Output 2 – comparison

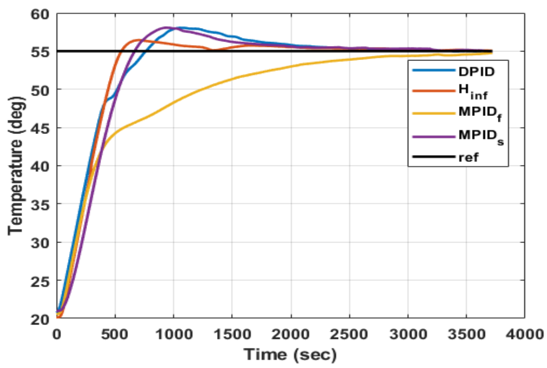

Fig. 16. Output 3 – comparison

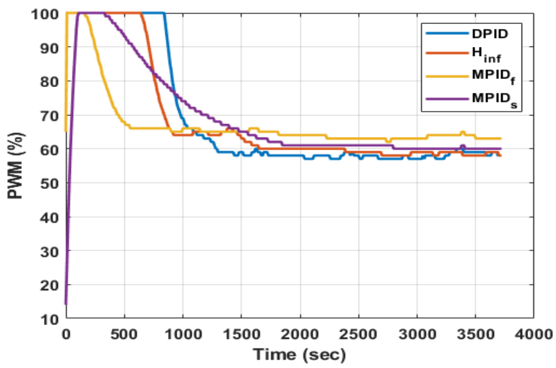

Fig. 17. Input 1 – comparison

For the seconds case the same reference change is applied at all loops. The amplitude is only 15 degrees, leading to smaller control signals and avoiding going into the saturation limit. Results are shown on Fig. 20 to Fig. 22. Notations are the same as in the previous case. Performance criteria are calculated and arranged in TABLE II.

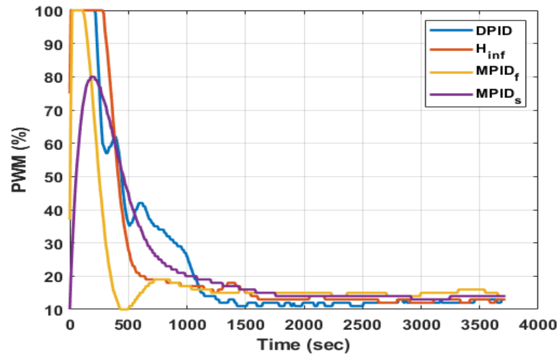

Fig. 18. Input 2 – comparison

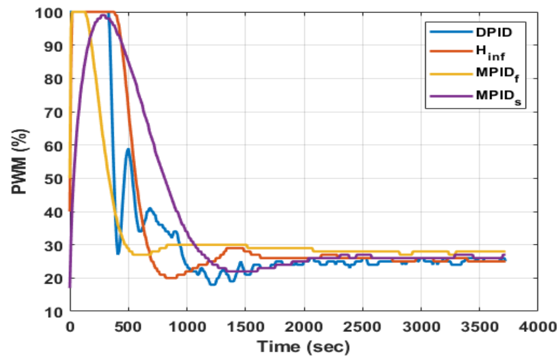

Fig. 19. Input 3 – comparison

The results above are in line with the frequency response of the singular values of the sensitivity functions of the closed loop system. All systems operate without entering saturation state. The “slow” PID leads to smaller control amplitudes at the expense of slow transient response. The “fast” PID leads to almost the same behavior as the full order H_∞ controller, which shows the potential of the fixed order design of a multivariable PID. Interesting enough is the result for the decentralized PID controller with the static decoupling. Excluding the settling time all other criteria are quite close to the rest of the stack, showing the possibility of a simple control strategy without the need of full knowledge of a model. The slow settling time might be due to not adequate anti wind-up mechanism.

V. Conclusion

The current paper investigated the viability of the *hinfstruct()* MATLAB function for design of multivariable discrete PID controllers. Two multivariable PIDs were designed and were compared against a full order H_∞ controller and a decentralized PID using static decoupling matrix. Frequency response of the singular values of the closed loop sensitivity functions were plotted. The controllers were tested on a laboratory thermal plant.

They were implemented using velocity form on a PLC controller. Two sets of test were conducted, using different amplitudes for the reference signal change, leading to different operating conditions of the closed loop systems. The results showed comparable performance between the full order H_∞ controller and the multivariable PIDs, supporting the idea of the viability of fixed order and fixed structure controller design.

TABLE I. PERFORMANCE INDEX CASE 1

Criterion / Type and order	Rise Time [s]			Settling time [s]			Overshoot [%]			Se [-]			Su [-]		
HPID fast	1661.6	1286.6	1167.1	2436.7	2691.7	2506.5	0	0	0	11.31	7.81	9.2	67.96	27.47	37.95
HPID slow	697.55	518.79	518.2	2010.4	2008.8	1563.7	3.52	3.54	5.57	11.0	7.58	8.91	70.37	28.87	44.46
H_∞	604.83	351.26	433.62	1057	988.8	892.45	2.61	2.9	2.63	10.3	6.51	8.13	70.41	34.53	43.54
DPID	595.13	367.16	502.54	1922.4	1963.2	1794.9	7.6	8.67	5.55	9.96	6.29	7.73	71.2	30.05	40.62

TABLE II. PERFORMANCE INDEX CASE 2

Criterion / Type and order	Rise Time [s]			Settling time [s]			Overshoot [%]			Se [-]			Su [-]		
HPID fast	1661.6	1286.6	1167.1	2436.7	2691.7	2506.5	0	0	0	11.31	7.81	9.2	67.96	27.47	37.95
HPID slow	697.55	518.79	518.2	2010.4	2008.8	1563.7	3.52	3.54	5.57	11.0	7.58	8.91	70.37	28.87	44.46
H_∞	604.83	351.26	433.62	1057	988.8	892.45	2.61	2.9	2.63	10.3	6.51	8.13	70.41	34.53	43.54
DPID	595.13	367.16	502.54	1922.4	1963.2	1794.9	7.6	8.67	5.55	9.96	6.29	7.73	71.2	30.05	40.62

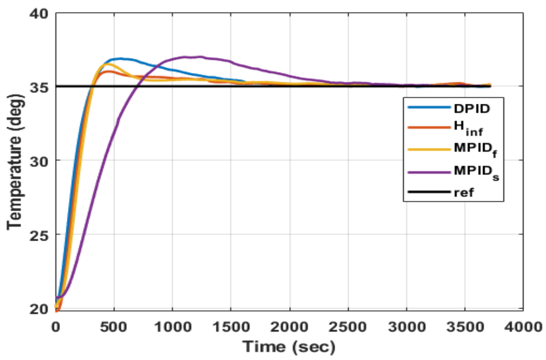

Fig. 20. Output 1 – comparison

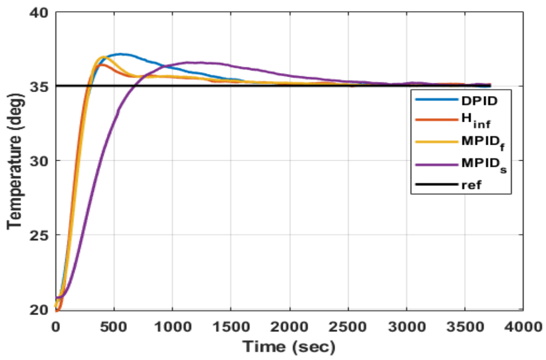

Fig. 21. Output 2 – comparison

Since the received result are just from a single plant, they should not be generalized, but they give a good insight to the corresponding problem. Potential future work might include the robustness topic of the closed loop systems, designed by use of fixed structure PID controllers. A new plant is already constructed – two water level coupled tanks with possibility to change the outlet valve positions, leading to variable dynamics.

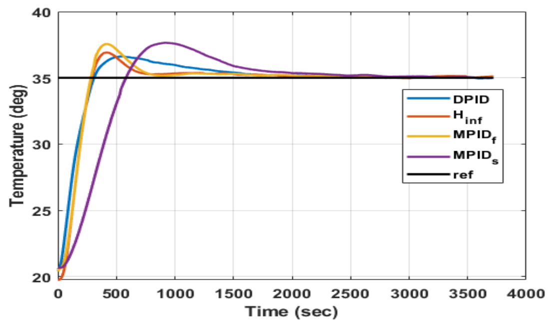

Fig. 22. Output 3 – comparison

ACKNOWLEDGMENT

This research is supported by the Bulgarian Ministry of Education and Science under the National Program "Young Scientists and PostdoctoralStudents - 2". Also the authors would like to thank the Research and Development Sector at the Technical University of Sofia for the financial support.

REFERENCES

[1] Ramon Vilanova, Antonio Visioli, "PID Control in the third Millennium", Springer 2012

[2] George Zames, "*Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximate inverses*", *IEEE Transactions on Automatic Control. 26,* page 301-320, 1981

[3] Lennart Ljung, "System Identification: Theory for the user 2nd edition", Prentice Hall 1999

[4] Da-Wei Gu, Petko H. Petkov and Mihail M. Konstantinov, "Robust Control Design with MATLAB", Springer, 2013

[5] Apkarian, Pierre, and Dominikus Noll. "Nonsmooth H_∞ Synthesis." *IEEE Transactions on Automatic Control*, 51, page 71-86

[6] MATLAB Online documentation for *hinfstruct()* function - https://www.mathworks.com/help/robust/ref/dynamicsystem.hinfstruct.html;jsessionid=754a098cc03b966668a0e9ede187

[7] Божидар Раков, "Независим синтез на ПИД регулатор за тримерен обект", Годишник на Технически Университет 2018

[8] Sigurd Skogestad, Ian Postlethwaite, "Multivariable Feedback Control: Analysis and Design 2nd Edition", Wiley & Sons 2001

[9] Karl Astrom, Tore Hagglund, "Advanced PID Control", Instrumentation, Systems, and Automation Society 2006