

Lukáš BARTALSKÝ*, Gabriel HULKÓ**

MODELING AND DESIGN OF MECHATRONIC CONTROL PREHEATING
FOUNDRY MOLD AS A SYSTEM WITH DISTRIBUTED PARAMETERS

MODELOVANIE A NÁVRH RIADENIA PREDOHREUV MECHATRONICKEJ
ZLIEVARENSKEJ FORMY AKO SYSTÉMU S ROZLOŽENÝMI PARAMETRAMI

Abstract

The article deals with modelling and control of preheating of mechatronics foundry mold as a system with distributed parameters. At the beginning of article the fundamental structures of distributed parameters systems are described. The mechatronics foundry mold and simulation model of it in COMSOL Multiphysics are described in the next part of article. Article also described the system identification of mold by using the process responses in MATLAB Identification Toolbox. Subsequently deals with the simulation control loop by using DPS Blockset for MATLAB & Simulink.

Abstrakt

Predkladaný článok rieši otázky modelovania a riadenia predohrevu mechatronickej zlievarenskej formy ako systému s rozloženými parametrami. Úvodom sú v skratke predstavené základné štruktúry systémov s rozloženými parametrami, ďalej je opísaná mechatronická zlievarenská forma a tvorba simulačného modelu v programovom prostredí COMSOL Multiphysics. Po identifikácii dynamických charakteristík je zostavený systém riadenia v softvérovom prostredí DPS Blockset pre MATLAB & Simulink.

Keywords

preheating foundry molds, system with distributed parameters, simulation, identification, control synthesis, DPS Blockset.

1 INTRODUCTION

The paper deals with controlled preheating of mechatronics foundry mold. Due to the shape of the experimental device, which was purposely designed in the shape of a cross, is required controlled preheating. On casting occurred mistakes, like shrinkages or depressions, without pre-heating before casting process. Nowadays tendencies in industry are save the costs of production, from this point of view is advantageous to use a controlled preheating compare to preheat the entire form to the same temperature. The main advantage of controlled preheating is that we can run it as a system with distributed parameters, so we can achieve a different required temperature in a given zones. Article

* Ing., Institute of Automation, Measurement and Applied Informatics, Faculty of Mechanical Engineering, Slovak University of Technology in Bratislava, Námestie Slobody 1, Bratislava, tel. (+421) 2 5249 7193, e-mail lukas.bartalsky@stuba.sk

** prof. Ing., DrSc., Institute of Automation, Measurement and Applied Informatics, Faculty of Mechanical Engineering, Slovak University of Technology in Bratislava, Námestie Slobody 1, Bratislava, tel. (+421) 2 5249 7193, e-mail gabriel.hulko@stuba.sk

deals with design of control for mechatronics foundry mold by using of software product - DPS Blockset.

2 SYSTEMS WITH DISTRIBUTED PARAMETERS

Technological processes having a continuum-based nature, are generally described by Partial Differential Equations (PDE) and in the sense of systems and control theory are understood as Distributed Parameter Systems (DPS). In most practical cases we introduce this description in a linearized neighborhood of the controlled system's steady state operating points thus obtaining a linear PDE. Distributed quantities from the right and left sides of PDE in input/output relation give Distributed input and Distributed parameter output Systems or simply Distributed input and Distributed output Systems (DDS), Fig. 1.



Fig. 1 Distributed Input and Distributed Output System.

- DDS – distributed input and distributed output system
- $U(\mathbf{x}, t) - U(x, y, z, t)$ – distributed input quantity
- $Y(\mathbf{x}, t) - Y(x, y, z, t)$ – distributed output quantity

For control purposes actuators with lumped parameter inputs and distributed parameter generators as well are associated to DDS and so we obtain controlled Lumped input and Distributed output Systems (LDS), Fig. 2 and Fig. 3.

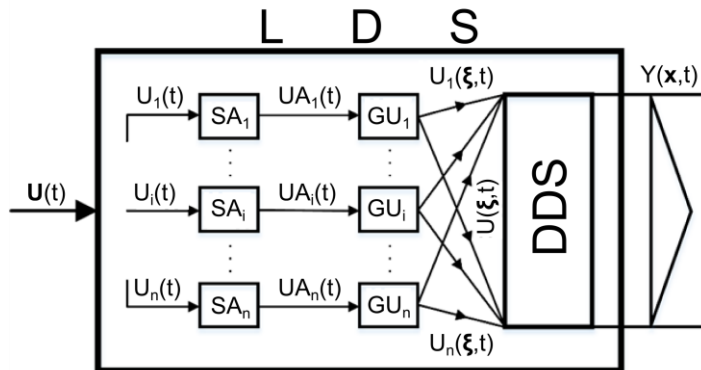


Fig. 2 Inner structure of Lumped Input and Distributed Output System.

- LDS – lumped input and distributed output system.
- $\{SA_i\}_i$ – actuators as lumped parameter subsystems.
- $\{GU_i\}_i$ – distributed quantity generators.
- DDS – distributed input and distributed output system.
- $U(t) = \{U_i(t)\}_i$ – lumped parameter input vector.
- $\{UA_i(t)\}_i$ – lumped input variables of $\{GU_i\}_i$.

- $\{U_i(\xi, t)\}_i$ – distributed output variables of $\{GU_i\}_i$.
- $U(\xi, t)$ – overall distributed input quantity of DDS.
- $Y(\mathbf{x}, t) = Y(x, y, z, t)$ – distributed output quantity.

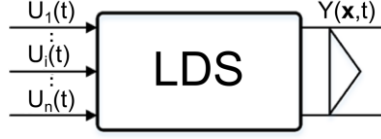


Fig. 3 Lumped input and distributed output system.

- LDS – lumped input and distributed output system
- $\{U_i(t)\}_i$ – lumped input quantities
- $Y(\mathbf{x}, t) = Y(x, y, z, t)$ – distributed output quantity

Assuming zero initial and boundary conditions, the system's input/output relation is given in continuous form as

$$Y(\mathbf{x}, t) = \sum_{i=1}^n Y_i(\mathbf{x}, t) = \sum_{i=1}^n \mathcal{G}_i(\mathbf{x}, t) \otimes U_i(t) = \sum_{i=1}^n \int_0^t \mathcal{G}_i(\mathbf{x}, t - \tau) U_i(\tau) d\tau \quad (2.1)$$

and in discrete form we have

$$Y(\mathbf{x}, k) = \sum_{i=1}^n Y_i(\mathbf{x}, k) = \sum_{i=1}^n \mathcal{G}H_i(\mathbf{x}, k) \oplus U_i(k) = \sum_{i=1}^n \sum_{q=0}^k \mathcal{G}H_i(\mathbf{x}, k - q) U_i(q) \quad (2.2)$$

where \otimes indicates the convolution product and \oplus the convolution sum, $\{U_i(t)\}_i$ and $\{U_i(k)\}_i$ are continuous and discrete lumped input variables, $Y(\mathbf{x}, t)$ and $Y(\mathbf{x}, k)$ are overall continuous and discrete distributed output quantities, $\{\mathcal{G}_i(\mathbf{x}, t)\}_i$ is the LDS distributed impulse response to the i -th lumped input, $\{\mathcal{G}H_i(\mathbf{x}, k)\}_i$ its discrete version assuming zero order hold units H: HLDS.

$$\{Y_i(\mathbf{x}, t)\}_i = \{\mathcal{G}_i(\mathbf{x}, t) \otimes U_i(t)\}_i = \left\{ \int_0^t \mathcal{G}_i(\mathbf{x}, t - \tau) U_i(\tau) d\tau \right\}_i \quad (2.3)$$

and

$$\{Y_i(\mathbf{x}, k)\}_i = \{\mathcal{G}H_i(\mathbf{x}, k) \oplus U_i(k)\}_i = \left\{ \sum_{q=0}^k \mathcal{G}H_i(\mathbf{x}, k - q) U_i(q) \right\}_i \quad (2.4)$$

are individual distributed output variables of LDS (first the continuous, latter the discrete) belonging to individual lumped inputs. [1]

Applying Laplace transform we obtain

$$Y(\mathbf{x}, s) = \sum_{i=1}^n Y_i(\mathbf{x}, s) = \sum_{i=1}^n S_i(\mathbf{x}, s) U_i(s) \quad (2.5)$$

$$\{Y_i(\mathbf{x}, s) = S_i(\mathbf{x}, s) U_i(s)\}_i \quad (2.6)$$

and after Z-transform we have

$$Y(\mathbf{x}, z) = \sum_{i=1}^n Y_i(\mathbf{x}, z) = \sum_{i=1}^n S_i(\mathbf{x}, z)U_i(z) \quad (2.7)$$

$$\{Y_i(\mathbf{x}, z) = S_i(\mathbf{x}, z)U_i(z)\}_i \quad (2.8)$$

3 EXPERIMENTAL MECHATRONIC FOUNDRY MOLDS

Experimental device which is used to investigate the possibility of controlled preheating of foundry mold and active cooling of casting was constructed on Institute of Automation, Measurement and Applied Informatics on Faculty of Mechanical Engineering in collaboration with Institute of Technology and Materials. It is equipped with sensors and actuating (power) technology to generate and simultaneous observation of the temperature field in the body of the mold. For control of sensing equipment a power electronics is used with MATLAB/Simulink, which is support the measurement and control in real time.

Cross joint is a common structural element, but in the casting technology appears in cross joint defects like shrinkages or depressions, due to thermal node placed in this area of casting. So it is a fundamental problem this technology, whose solution is suggested in this article. Shape of casting cross joint can be seen in the Fig. 4.

In this construction arrangement of arms of the foundry mold are used a different cross-section to modelling of synchronous cooling of mold arms. And finally is in the solution considered with large distance between the temperature node and the tang (source of liquid material) of the foundry mold. Due of the long distance arised problem with putting liquid metal to solidifying casting.

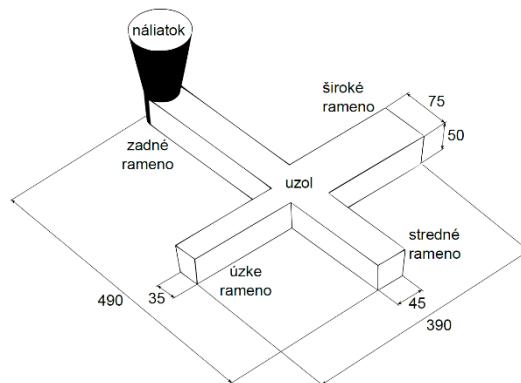


Fig. 4 The shape and dimensions of the model casting

The final design of casting emerged as a compromise taking into account the possibility of foundries of Institute of Technology (ÚTM FME), the cost of the device production and making the experiments on ready device. The design and size of entire device is derived from the shape and size of finite casting products (cross joint). The main part of entire device is steel mold with integrated measuring and actuating parts.

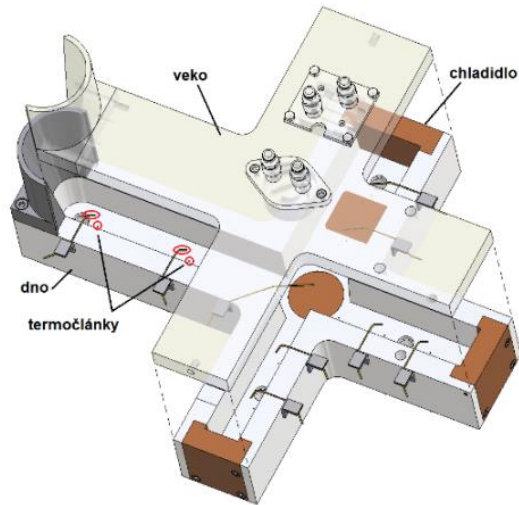


Fig. 5 Steel mold with integrated measuring and actuating parts.

3.1 Technology of mold preheating

Heating circuit

Heating power required for preheating process, provides 26 integrated electric bodies, every has nominal power 400W, divided into five zones, Fig. 5 a Fig. 6:

ZONE 1 Narrow arm (35mm)	1600W
ZONE 2 Cross joint	2400W
ZONE Wide arm	1600W
ZONE4 Medium arm (45mm)	1600W
ZONE4 Long arm with mold tang	1600W

Entire installed power is 10,4kW. The temperature of the mold is measured by eleven thermocouples (TE-P01 to TE-P11). Amount and power of heating patrons is designed by taking into account the requirements of rate of achieve the desired temperature and the knowledge about significant capacitive delay. [2].

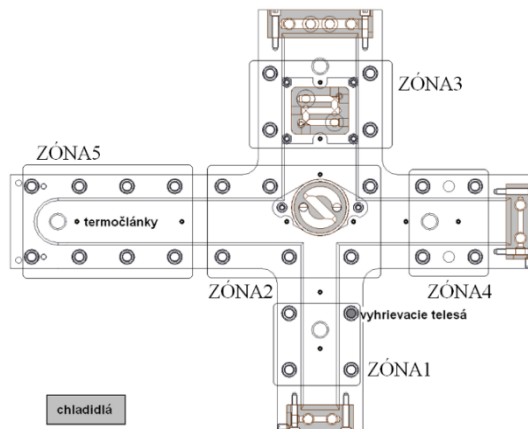


Fig. 6.Heating circuit

4 MODELING IN COMSOL MULTIPHYSIC

Preheating of mechatronics casting mold was modeled by using software product COMSOL Multiphysics. Geometrical CAD model of mold was imported into COMSOL. Subsequently was defined boundary conditions and ambient temperature on 23°C. Heating zones on comsol model was defined by the zones (ZONE no.1 - ZONE no.5) on real casting mold. After that on comsol model of a mold was created a mesh of finite elements.Fig. 7.

Experimental way was found power ratio of heating patrons to hold steady temperature field, which include the operating temperature. Operating temperature of the preheated casting mold is determined to be 250°C. Based on this knowledge, simulation of step responses on individual zones were performed.

Unit step on each zone has 20% of nominal zone power. Overview of power ratio before unit step and with unit step is in table Table 4.1. The Matrices of transient responses with dimension 11x3601 were results of simulations. The matrices have 11 rows because of 11 measuring points. Simulation time were 18000 seconds, sampling time were 5 seconds and the first sample give us 3601 columns of matrices. Simulation time were proposed by assumption about duration of preheating process and duration of measuring of it, with real sampling time used by measuring. Results of simulation after rising up of power on the first zone are in the Fig. 9 up to Fig. 10. In Fig. 8 can be see steady temperature field of casting mold at the end of simulation, increase of power at the first zone is colored yellow. In the Fig. 9 can be see steady temperature in measuring points inside the casting mold. On the picture is an obvious increase of temperature at the first zone compared to other zones. Transient responses after increase the power at the first zone are in the Fig. 10. In this figure can be seen that the varying of the power at one zone can be effect on each other zone of casting mold. Because of this, is required has transient responses in the form of matrices by dimension 11x3601. These matrices can be used for correct control design of preheating by using the DPS Blockset. [3].

Table 4.1 Heating power values before and after increase of power

Zone	Original heating power	20% increase of heating power
1	485	582
2	500	600
3	650	780
4	465	558
5	715	858

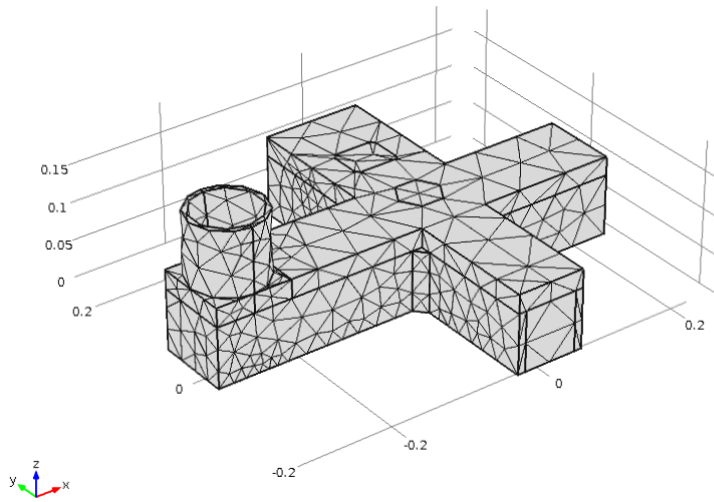


Fig. 7 Comsol model of casting mold - finite element mesh

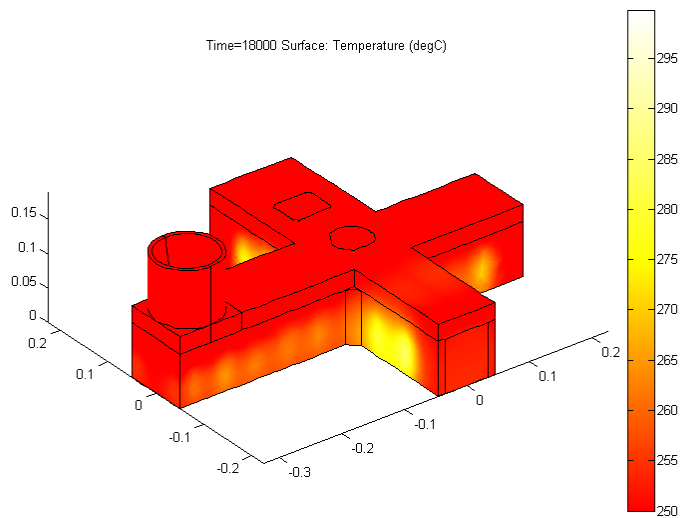


Fig. 8 Steady temperature field of casting mold.

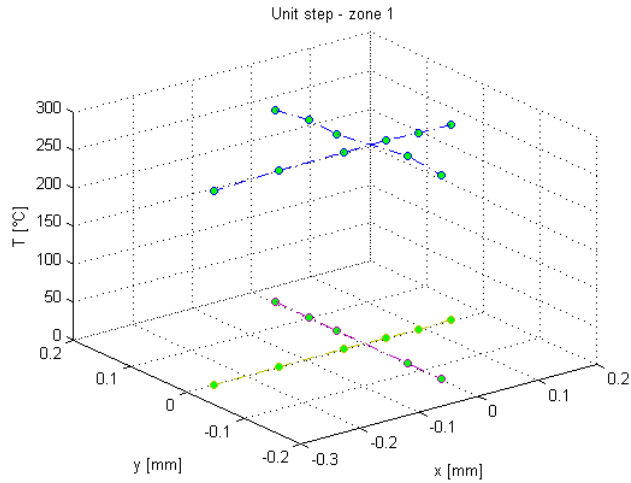


Fig. 9 Steady temperature at measuring point

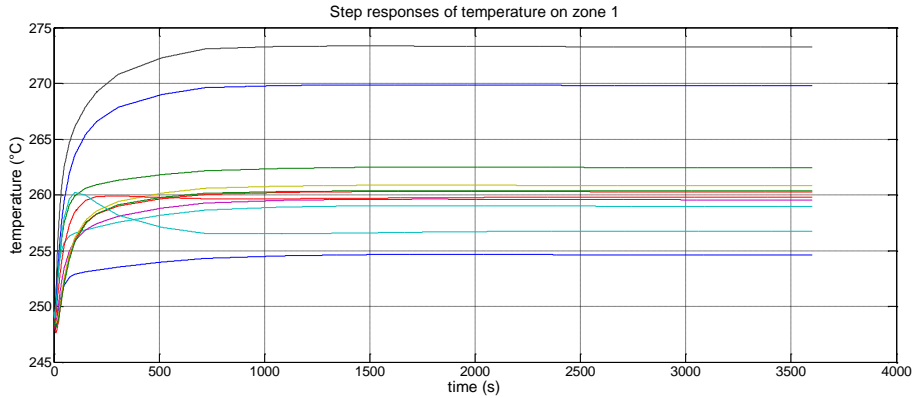


Fig. 10 Development of temperature in measuring point after power up at ZONE no.1.

5. IDENTIFICATION IN MATLAB

After simulation were data from Comsol imported to MATLAB by using the LiveLink from Comsol to MATLAB. Transient functions were identified in MATLAB by using the Identification Toolbox, Fig. 11. For the identification process, from each matrices of transient function were selected transient function with the fastest dynamics. Transient function with fastest dynamics was identified in Identification Toolbox.

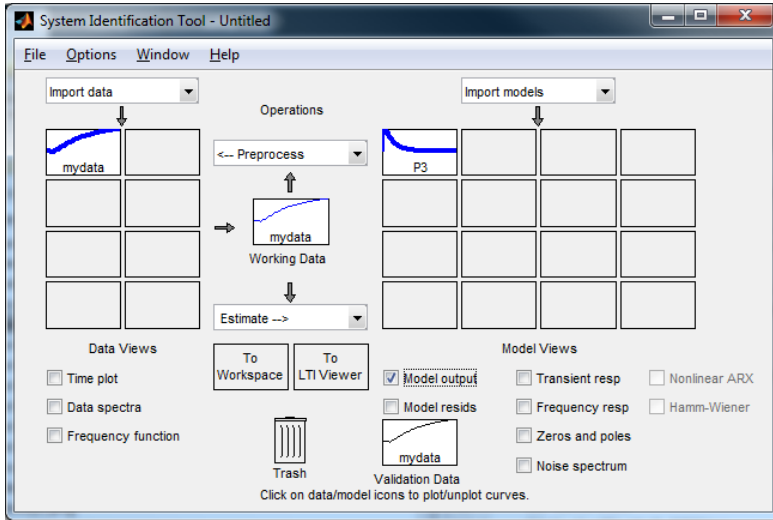


Fig. 11 Identification Toolbox

The Fig. 12 show us loading of transient characteristics. To loading of this characteristics is required define the input signal, which caused the transient response. Our input signal was unit step, whose value was 20% increase to heating power compare to it's original value (table 4.1).

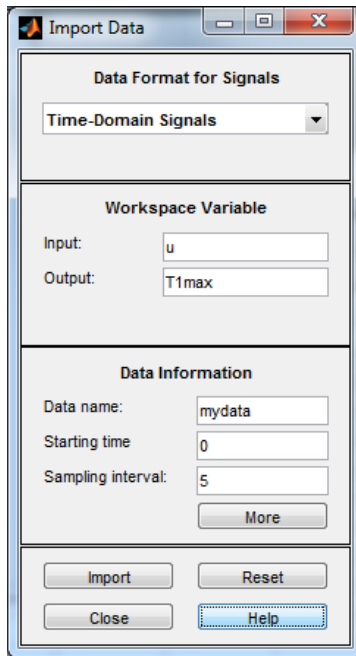


Fig. 12 Import identification toolbox

The task of identification is find coefficients of transfer function of system. The system is represented by transient characteristics obtained by simulation in Comsol Multiphysics. Transfer function is in form (5.1).

$$\frac{K}{(1 + T_{p1}s)} \tag{5.1}$$

Fig. 13 show setting for our transfer function coefficient estimation. On Fig. 14 can be seen, that parameters estimation of model is almost identical to our model. Symmetry between these two models is 97,6%.[3].

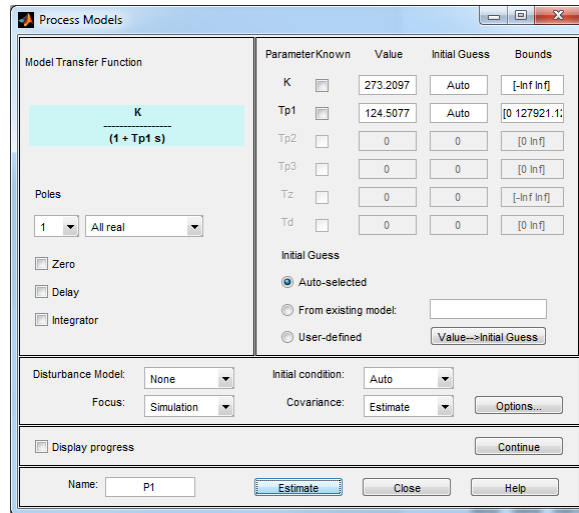


Fig. 13 The settings for parameters estimation and estimated parameters.

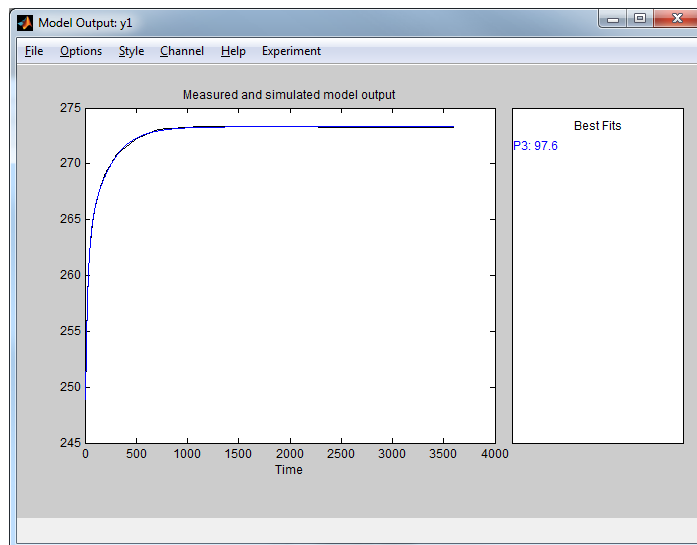


Fig. 14 Comparison of the original and the estimated model.

6. CONTROL DESIGN BY USING DPS BLOCKSET

The control synthesis was performed in the virtual environment of MATLAB&Simulink by using the third party software product DPS Blockset. DPS Blockset je specific software product developed at Institute of Automation, Measurement and Applied Informatics. DPS Blockset is fundamental for synthesis of systems with distributed parameters. The main part of DPS Blockset is HLDS (DPS Blockset Simulink's) block. HLDS block contains an array of transient characteristics from individual zones. DPS Space Synthesis is another important block. It's handle with space with

space synthesis. Desired value is a vector by 11×1 and the HLDS output has dimension 11×1 . So in this case we need space synthesis, because of 5 PI controller in time synthesis. Space synthesis block processes control error to required form for time synthesis.

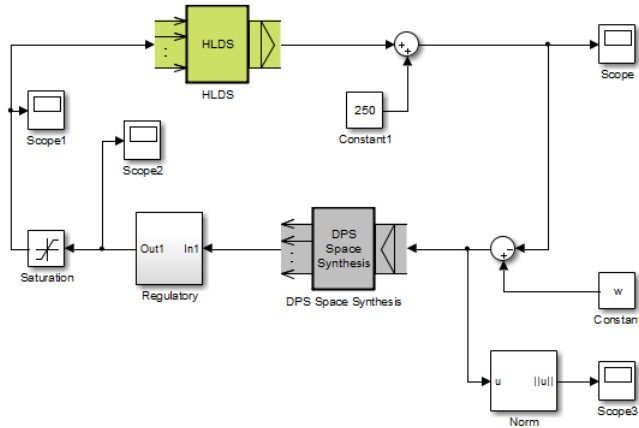


Fig. 15 Control scheme with DPS Toolbox

HLDS block represents controlled system, in our case it's casting mold, as can be see in Fig. 15. HLDS block in Fig. 15 contains transient characteristics obtained from Comsol simulation, so this block fully represents real device. These transient characteristics from Comsol simulation inside a HLDS block (in Fig. 15) causes that, HLDS representing casting mold gives by simulation in MATLAB the same responses like a real casting mold in lab. In control loop is added constant by Simulink's constant block by 250, it's because of setting on temperature operating point 250°C . Meaning of added constant is that, the transient characteristics obtained in HLDS block started in the beginning of grid system. Because of this HLDS block feature the constant block was used. Constant representing operating point shift the beginning of grid system of transient characteristics contained in HLDS block. Without constant added to control loop controllers wouldn't work properly. In control loop in Fig. 15 was used the saturation block too. It's handle with saturation of controllers output signals. It's necessary to cut controllers output, because of operating with heating elements. Control of heating elements works with voltage signal by 0-10V.

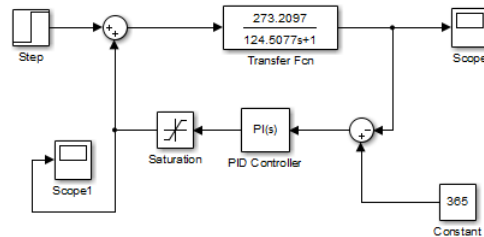


Fig. 16 Tuning scheme of PI controllers parameters

Tunning of controllers parameters has been done by scheme in the Fig. 16. The Transfer function block contained parameters, obtained by using of Identification toolbox. Tunnig of P- and I-component (P-proportional, I-integration) of controllers has been done for individually device zones separately. Setting up controllers parameters has been done automatically by using of PID Controller's block function PID Tune. This function offer to matlab user the possibility change these estimated parameters manually, if the estimated parameters aren't satisfying. The user has to know

fundamentals of control theory for correct setting up of controllers parameters. The output temperature responses by using tuned controllers can be see in the Fig. 17.

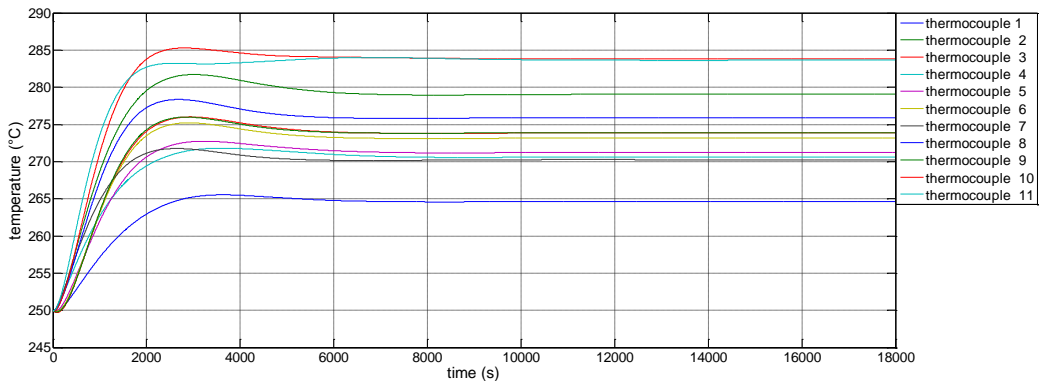


Fig. 17 Output temperature by using of controllers.

Required temperature field was given by vector w , this is a vector of desired temperature on thermocouples on casting mold. Reached steady temperature is about desired value of output temperature. It's meets the requirement of technology[4].

$$w = [270 \ 270 \ 275 \ 270 \ 270 \ 275 \ 270 \ 270 \ 275 \ 280 \ 285]^\circ\text{C}.$$

7. CONCLUSION

The article describes how the design of mechatronic control preheating foundry mold as a system with distributed parameters. Currently is casting mold controlled in real time by using of measurement cards in the program MATLAB & Simulink. Currently it's working on control of casting mold by PLC system. The advantage of PLC system to measurement card in control is better usability in the industry and affordability. It's in plan for future that PLC completely relace measurement cards and desktop PC in control of preheating or cooling of casting mold system.

ACKNOWLEDGMENT

The author would like to thank APVV, project ID: APVV-0131-10 “*High-tech solutions for technological processes and mechatronic components as controlled distributed parameters system*”. Next ASFEU in framework of projects ITMS-26240220072 “*Center of Competence for intelligent technologies and informatization of system and services*” and project ITMS-26240220084 “*University scientific park STU in Bratislava*”. This support is very gratefully acknowledged.

REFERENCES

- [1] HULKÓ, G. a kol. Modelovanie, riadenie a návrh systémov s rozloženými parametrami. Vydavateľstvo STU v Bratislave, ISBN 80-227-1052-0. 1998.
- [2] NOGA, P.: Modelovanie a riadenie procesov zlievania ako systémov s rozloženými parametrami. Doktorandská dizertačná práca, Bratislava, 2011.
- [3] System Identification Toolbox : User’s Guid, www.mathworks.com, 2014
- [4] HULKÓ, G. et al.: Distributed Parameter Systems Blockset for MATLAB & Simulink. DPS Blockset for MATLAB & Simulink. Program CONNECTIONS of The MathWorks - partner produkt spoločnosti The MathWork. Bratislava – Natick, 2003-2012. www.mathworks.com/products/connections/product_detail/product_35800.html