

# LABREG SOFTWARE FOR IDENTIFICATION AND CONTROL OF REAL PROCESSES IN MATLAB

*Slavomír Kajan and Mária Hypiúsová*

Institute of Control and Industrial Informatics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, Slovak Republic

## Abstract

**This paper deals with software LABREG for identification and control of real processes. The LABREG operates in MATLAB environment using toolboxes SIMULINK, Ident, Control and Real Time. The interconnection of the controlled process with computer (PC) and program LABREG is used Advantech data acquisition card of type PCI 1711. The program LABREG provides a good tool for identification and dynamic process control in MATLAB and it was tested on the laboratory models of DC-motor, heat and water tank systems.**

## 1 Introduction

For the improvement of teaching subjects related to control theory of dynamic processes it is necessary to realise identification and PID controllers design not only using the simulation, but on real laboratory processes too [1]. Continuous process control using PC A/D board requires application software which is suitable for teaching [6].

This paper deals with software LABREG, which operates in MATLAB. MATLAB with toolboxes SIMULINK, Ident, Control and Real Time provides very good environment for the creation of application software LABREG [3, 4, 5, 7].

The program LABREG allows realising the following tasks:

- Measurement, monitoring and archiving of measured data of control loops
- Identification and obtaining of process model from step response and measured data
- PID controller design using process model
- Process control in two control loops with PID controllers of adjustable structure
- Analysis of control performance.

## 2 Interconnection of measurement system

The interconnection of the controlled process with computer (PC) and program LABREG is depicted in Fig. 1. The process is connected to a computer using Advantech data acquisition card of type PCI 1711. Its outputs (out1, out2) attend on process control with two control PID loops. Process outputs are measured by board inputs in1 to in4, where inputs in1 and in2 are used in control loops and other only for requirements of monitoring some process parameters. The interconnection of the controlled process with computer (PC) and program LABREG is depicted in Figure 1.

Consider the laboratory model of a DC-motor, shown in Figure 2. This model consists of two co-operating real DC servomotors, where the first one is connected as a motor and the other one as a generator. The mechanical interconnection is realised with inertia load and spring. Power supply, the signals measurement and motor control is supplied by motor electronics. In electronics, there is a RC component connected to the input of motor to enable the change of time constant and gain of controlled system. System dynamics parameters can be tuned with potentiometer. The value of voltage on potentiometer is measured as variable called  $U_{par}$ . Input variable  $U_{in}$  of motor has range between 0 – 10 [V]. Output variable  $U_{out}$  of motor is velocity, which has range between 0 – 10 [V].

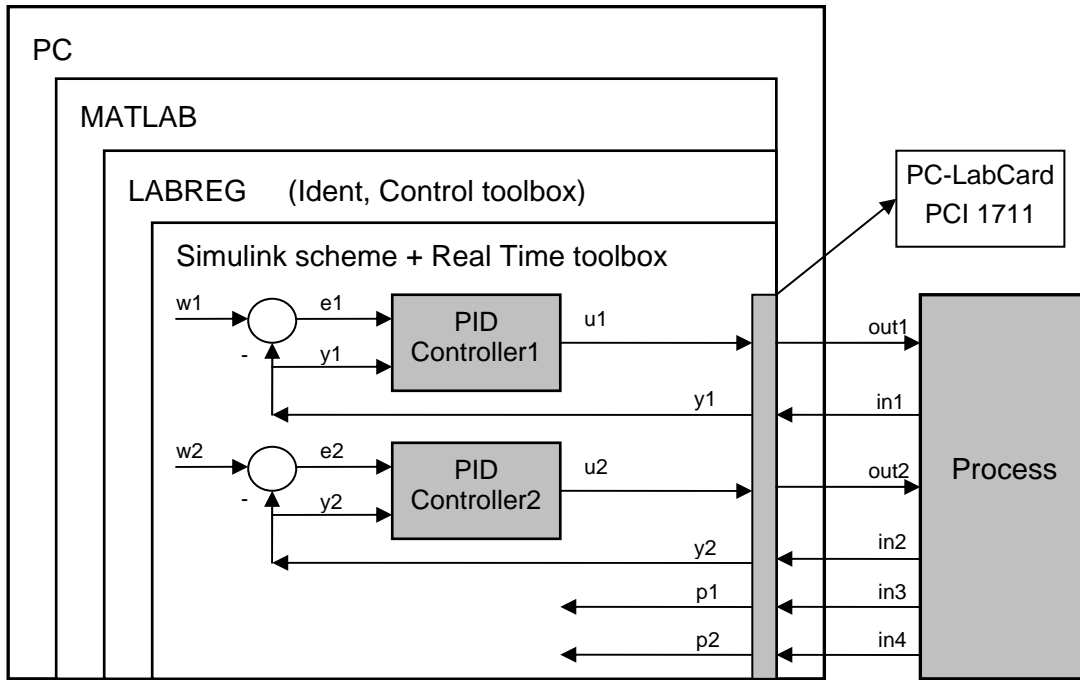


Figure 1: The interconnection of the controlled process with computer (PC) and program LABREG

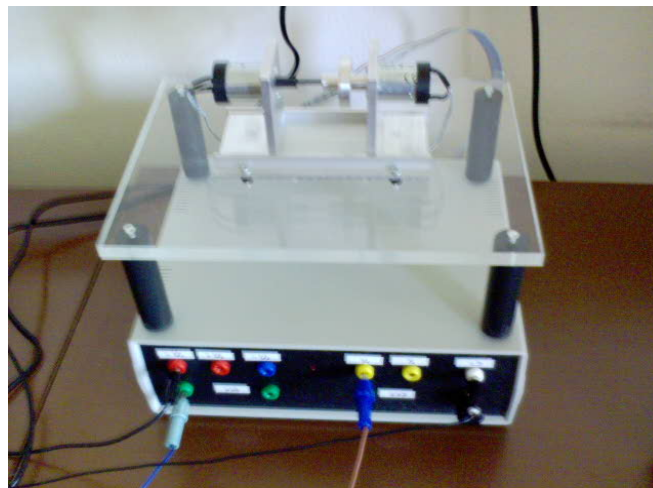
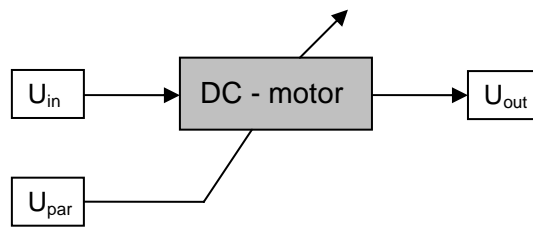


Figure 2: The laboratory model of DC motor with principle scheme

### 3 Structure of program LABREG

Program LABREG is started by command *Labreg\_start*, which initialises variables and I/O card. After initialising, the main program window is opened (Figure 3). From this window it is possible to start next windows for analysis, system identification, design of controller parameters, display of measured data, etc. In main window all program parameters are set. The principle block scheme of program LABREG is depicted in Figure 4.

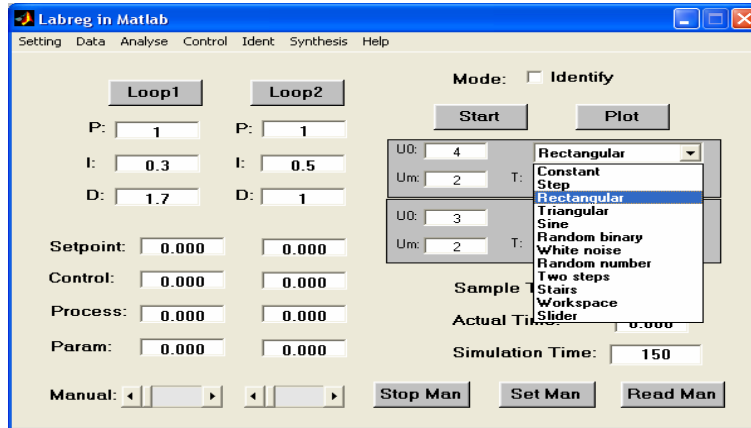


Figure 3: The main window of program LABREG

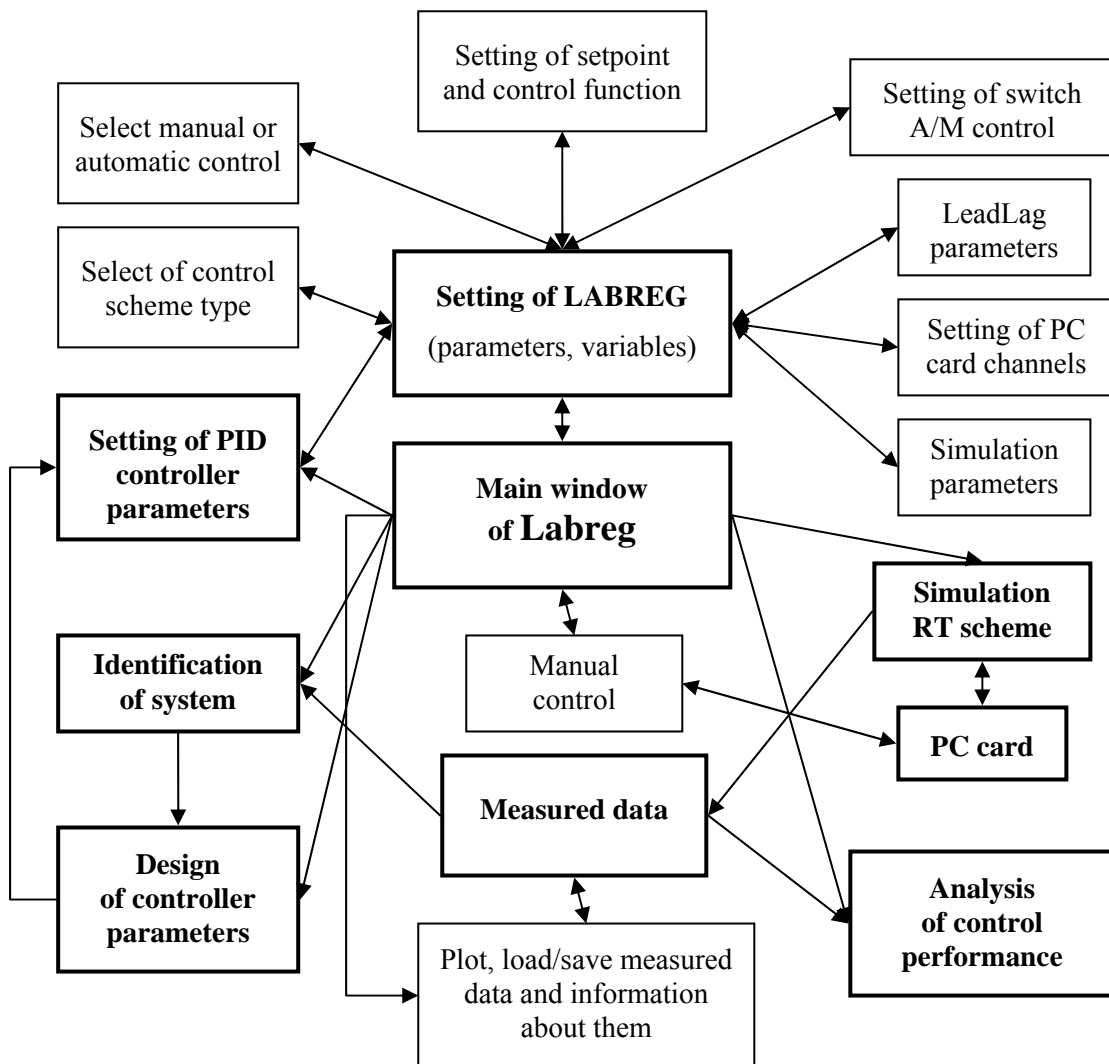


Figure 4: The principle block scheme of program LABREG

### 4 Measurement and control of process in real time

In this section quality of measurements in the process is concerned. When all program parameters in Main Window are prepared, it is possible to begin measurement by click on Start button. The simulation real time (RT) scheme for process control is opened (see Figure 5).

In the scheme, there are two control-loops for process control with two inputs and two outputs and other signals are only for requirements of monitoring some process parameters. The simulation scheme contains masked subsystems of controller, input functions and system, where user can set parameters of individual blocks. All these parameters can be set also in main window (Figure 3).

The windows for PID controller parameters setting in given loop and input functions setting for setpoint or manipulated variable are depicted in Figure 6.

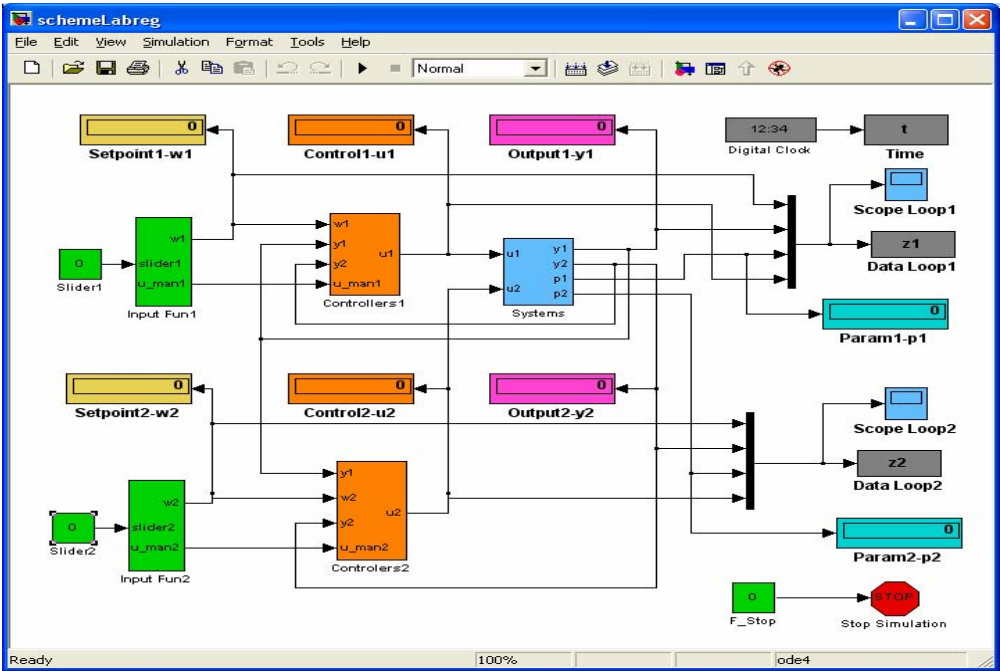


Figure 5: The simulation RT scheme for process control

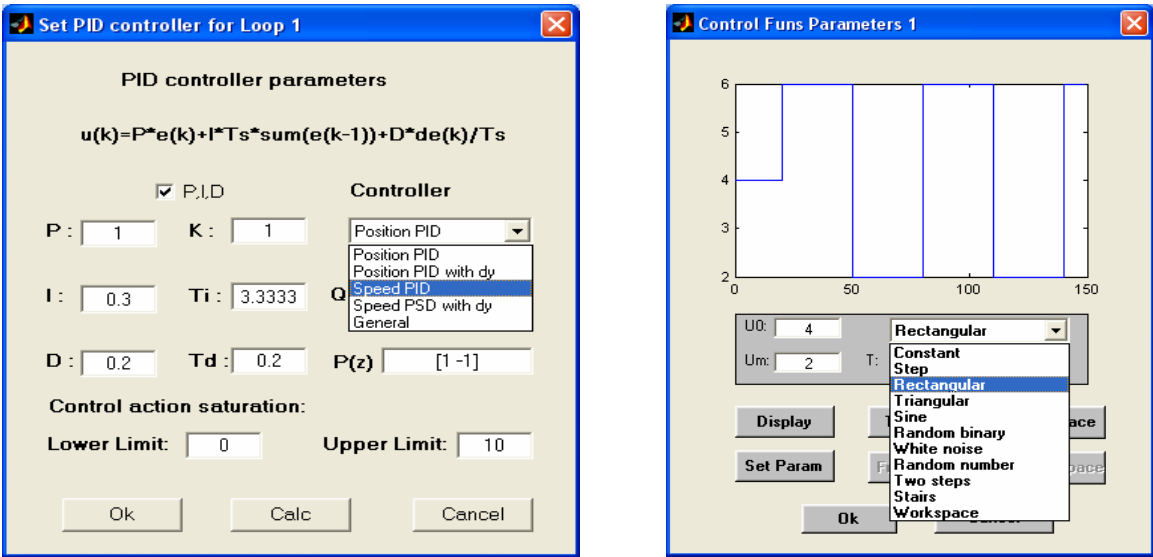


Figure 6: The setting windows for controller and input functions

## 5 System identification

When the measurements are finished, the data measured are saved and consequently, it is allowed to display data and to analyse control performance for closed loop, or to identify for open loop. The window for identification (see Figures 7 and 9) is started by click on Ident in main window menu. The system identification by program LABREG is possible using Autoregressive model with exogenous variable (ARX) method (1 – 5 order) [7] and Inverse dynamics (1 and 2 order with time delay) [2].

Identification program is realised for identification in process working point, i.e. for defined  $u_0$  and  $y_0$  about which the measured data are shifted into zero. Therefore it is necessary to use only a part of measured data, in which process transient response to achieve working point is not included, as illustrated in Figures 7 and 9 (part of Identification window).

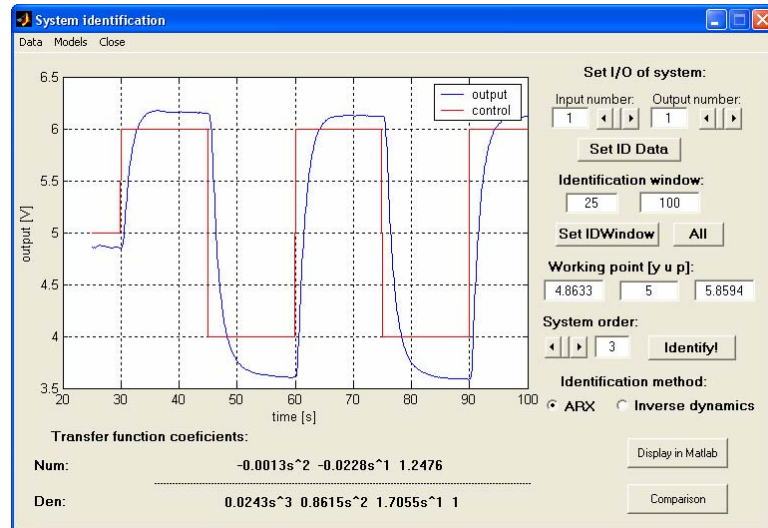


Figure 7: The system identification window of program LABREG, (identification with ARX model)

After the selection of identification method, identification window and system order is allowed, realise identification itself by click on ‘Identify!’ button. Identification results using ARX method are depicted in Figure 7 and comparison of time responses of ARX model output and process output is in Figure 8.

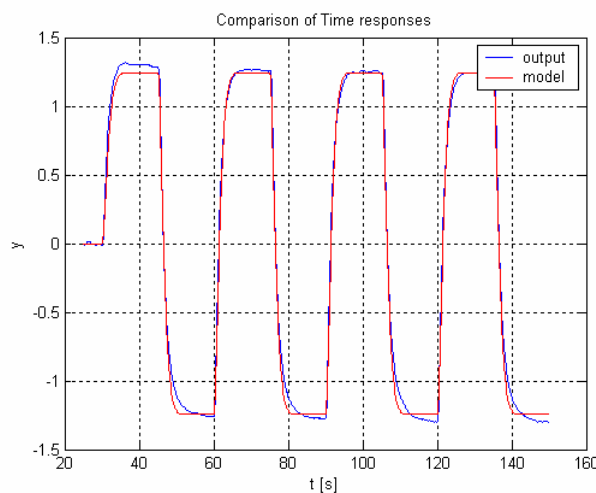


Figure 8: Comparison of time responses of ARX model output and process output

Identification results using Inverse dynamic method are depicted in Figure 9 and comparison of time responses of Inverse dynamic model output and process output is in Figure 10.

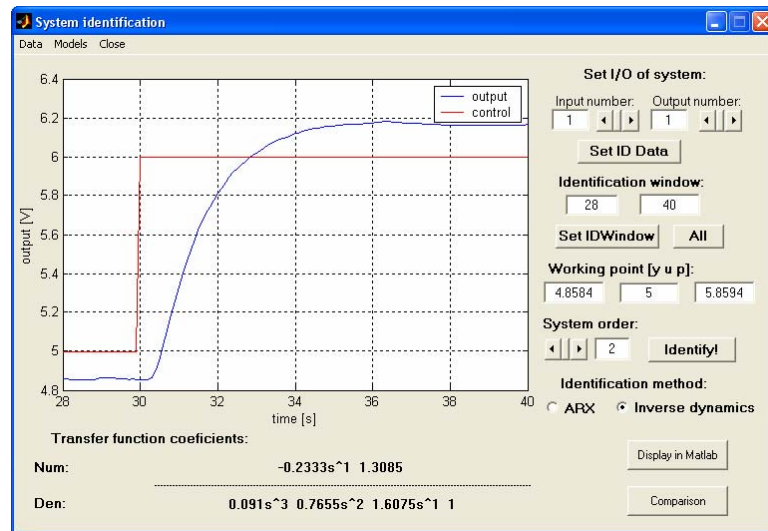


Figure 9: The system identification window of program LABREG, (identification with Inverse dynamic)

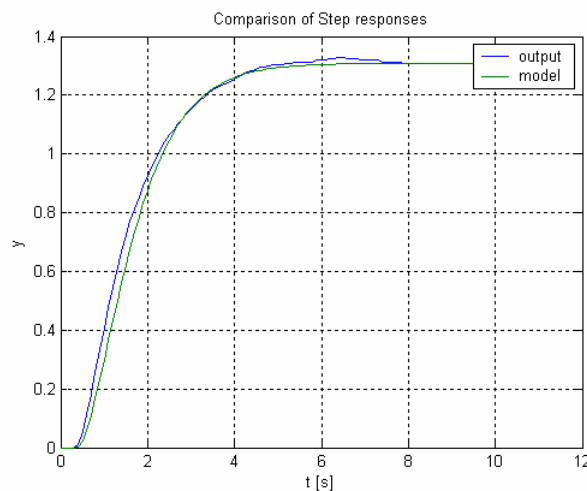


Figure 10: Comparison of step responses of Inverse dynamic model output and process output

## 6 Design of controller parameters

Controller parameters design on the basis of transfer function obtained from identification is started by click on *Synthesis* in main window menu. Thereafter the window for design of PID controller is opened as illustrated in Figures 11 and 12. For controller design it is necessary to select following items:

- Model type (ARX model, Inverse dynamic)
- Method of controller design (Optimal module, Sum of Time constant method (1-normal, 2-fast), method of Inverse dynamic)
- Controller type (PI, PD a PID)

Generally, PID controller is used in position and speed form with possible substitution of control error derivation ( $de$ ) by controlled variable derivation ( $dy$ ) in derivation part of controller. The position form of controller and its parameters are

$$G_R(s) = P + \frac{I}{s} + D \cdot s = K \left( 1 + \frac{1}{T_i \cdot s} + T_d \cdot s \right) \text{ then } u(t) = P \cdot e(t) + I \int e(t) dt + D \frac{de(t)}{dt} \quad (1)$$

and the speed form of controller and its parameters are

$$G_R(s) = P.s + I + D.s^2 = K \left( s + \frac{1}{T_i} + T_d.s^2 \right) \text{ then } \frac{du(t)}{dt} = P.\frac{de(t)}{dt} + I.e(t) + D\frac{d^2e(t)}{dt^2} \quad (2)$$

where  $P$  and  $K$  – gain of the controller

$I$  – integration part of controller

$D$  – derivative part of controller

$T_i$  – integration time constant of controller

$T_d$  – derivative time constant of controller

$u(t)$  – manipulated variable

$e(t)$  – control error

The calculation of PID controller parameters is started by click on *Design* button and the parameters are written to edit boxes. The results of controller parameters design in 1<sup>st</sup> loop used ARX model and Sum of Time constant method are depicted in Figure 11 and for Inverse dynamic model and Inverse dynamic method synthesis are in Figure 12.

Figure 11: The design of controller parameters window of program LABREG (for the design of PID controller in 1<sup>st</sup> loop used ARX model and Sum of Time constant method)

Figure 12: The design of controller parameters window of program LABREG (for the design of PID controller in 1<sup>st</sup> loop used Inverse dynamic model method too)

## 7 Measurement results

Control performance was verified by measurement for designed parameters of PID controller (see chapter 6). As an input of closed loop, steps in neighbourhood of working point were used. Measured data for PID controller designed by inverse dynamic method (from Figure 12) are depicted in Figure 13. From these data control performance was evaluated as is illustrated in Figure 14.

Control performance was evaluated also for closed loop with PID controller designed by Sum of Time constant method (from Figure 11) and the results are depicted in Figure 15.

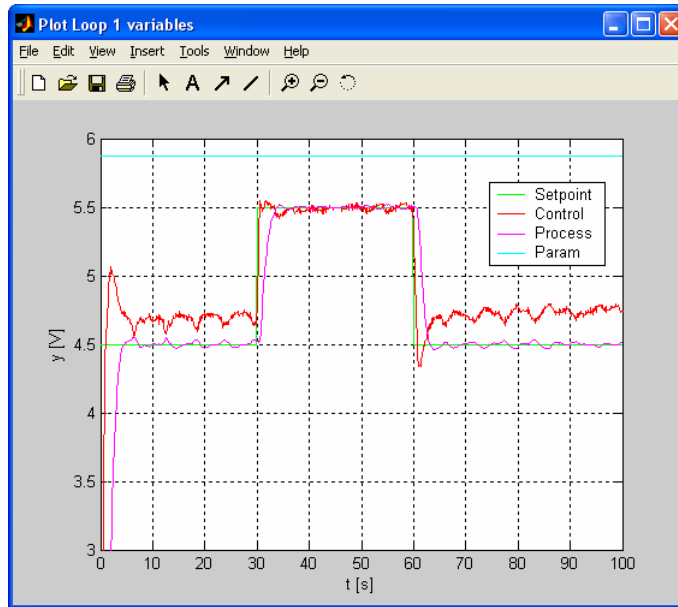


Figure 13: Measured data for PID controller designed by Inverse dynamic method

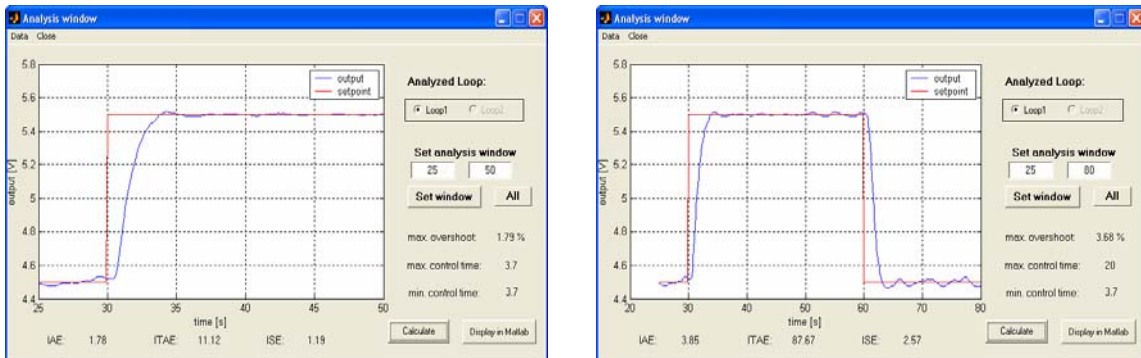


Figure 14: Analysis of control performance for PID controller designed by Inverse dynamic method

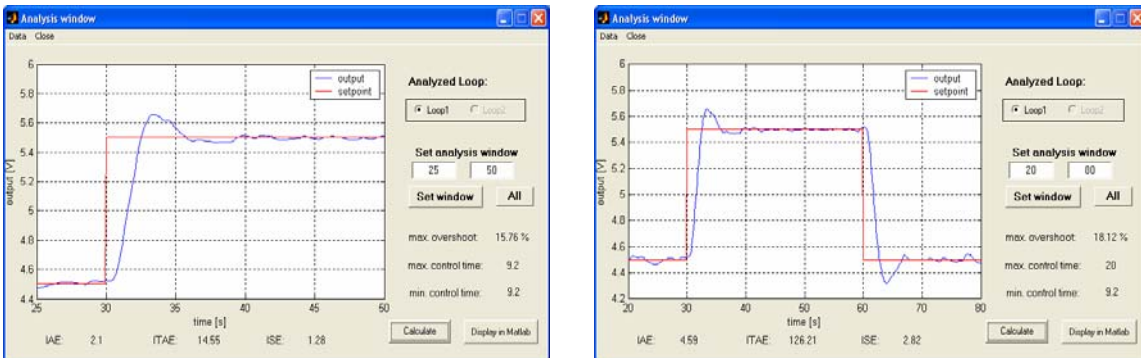


Figure 15: Analysis of control performance for PID controller designed by Sum of Time constant method



## 8 Conclusion

The main aim of this paper has been to present the new software LABREG for identification and control of real processes in MATLAB. This program was tested on the laboratory models of DC-motor, heat and water tank systems. The presented program provides a good tool for identification and dynamic process control in MATLAB and has improved teaching subjects related to control theory of dynamic processes where it is necessary to realise identification and PID controllers design not only using the simulation, but on real laboratory processes too.

## References

- [1] Š. Kozák. *Lineárne číslicové systémy I*, STU Bratislava, 1993.
- [2] M. Vítečková. *Seřízení regulátorů metodou inverze dynamiky*, Vysoká Škola Baňská - Technická Univerzita Ostrava, ISBN 80-7078-628-0, 1998
- [3] S. Kajan, Š. Kozák. *Matlab Simulink1*, STU Bratislava, ISBN 80-227-1213-2, 1999
- [4] The Humusoft s.r.o. *Real Time Toolbox version 3.12, user's manual*, 2003
- [5] The Mathworks. *Matlab ver. 6.5 (R13), user documentation*, 2002
- [6] I. Sekaj. *LABREG 3.1, Používateľská príručka programu*, KASR FEI STU, Bratislava, 1995
- [7] The Mathworks. *Identification System Toolbox, user documentation*, 2003

## Acknowledgments

The work has been supported by the grant agency VEGA No. 1/3100/06.

---

Ing. Slavomír Kajan, PhD.:

Institute of Control and Industrial Informatics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, Slovak Republic, Ilkovičova 3, 812 19 Bratislava, E-mail: slavomir.kajan@stuba.sk

Ing. Mária Hypiúsová, PhD.:

Institute of Control and Industrial Informatics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, Slovak Republic, Ilkovičova 3, 812 19 Bratislava, E-mail: maria.hypiusova@stuba.sk