U.P.B. Sci. Bull., Series C, Vol. 70, No. 3, 2008 ISSN 1454-234x

DESIGN PROCEDURE FOR INVERSE MODEL COMMAND: CONTROL METHOD FOR NONLINEAR PROCESSES

Ciprian LUPU[1], Cătălin PETRESCU[2], Alexandru ŢICLEA[3], Cătălin DIMON[4], Andreea UDREA[5], Bogdan IRIMIA[6]

Structurile cu model invers reprezintă una din soluţiile de succes pentru sistemele de timp real utilizate în reglarea proceselor neliniare. Utilizarea acestora impune rezolvarea unor probleme specifice cum sunt cele legate de determinarea caracteristicii statice a procesului, construcţia modelului invers sau proiectarea robustă a comenzii. Lucrarea propune o structură de reglare bazată pe modelul invers al procesului precum şi o metodologie practică originală corespunzătoare proiectării şi implementării fizice a acesteia. Aplicabilitatea acestei structuri este demonstrată utilizând o structură de reglare de tip RST. În final, sunt prezentate implementarea software şi rezultatele obţinute.

Structures with inverse model represent one of the successful solutions for the real-time control of the nonlinear processes. The use of these structures imposes solving some specific problems, like determination of static characteristic of the process, construction of inverse model or robust control law design. The paper proposes a structure and the correspondent original methodology of designing and physically implementation based on inverse model command. The applicability of the structure is proved using a real-time structure with an RST control algorithm. In the end, its software implementation and the obtained results are also shown.

Keywords: control systems, inverse model, robustness, real-time systems

1. Introduction

The essential condition for the real-time function of a control system is preserving the closed-loop performances in case of non-linearity, structural

---

[1] Reader, Dept. of Automatic Control and Computer Science, University "Politehnica" of Bucharest, Romania, cip@indinf.pub.ro
[2] Reader, Dept. of Automatic Control and Computer Science, University "Politehnica" of Bucharest, Romania, catalin@indinf.pub.ro
[3] Lecturer, Dept. of Automatic Control and Computer Science, University "Politehnica" of Bucharest, Romania, ticleaa@ yahoo.com
[4] Assist., Dept. of Automatic Control and Computer Science, University "Politehnica" of Bucharest, Romania, catalin.dimon@gmail.com
[5] Assist., Department of Automatic Control and Computer Science, University "Politehnica" of Bucharest, Romania, udrea.andreea@yahoo.com
[6] Assist., Dept. of Automatic Control and Computer Science, University "Politehnica" of Bucharest, Romania, irimiab@zappmobile.ro

disturbances or process uncertainties. A valuable way to solve these problems is inverse model command or also named direct command.

About inverse model numerous ancient and recent papers and researches exists. Few of this, with a very fortuity choosing procedure, can be mentioned: [12], [10], [1] etc.

In these researches there are a lot of inverse model proposed structures. According to them, the paper proposes a very simple and efficient structure presented in Fig. 1. This solution supposes adding of two commands: the first one "a direct command" generated by inverse model command generator, and the second generated by a classic and very simple algorithm (PID, RST etc.).

The first command, based on process static characteristics, is dependent on set point value and is designed to generate a corresponding value to drive the process's output close to imposed set point. The second (classic) algorithm generate a command that, corrects the difference caused by external disturbances and according to set point, by eventual bias error caused by mismatches between calculated inverse process characteristic and situation from real process.

Presented solution proposes treating of these inverse model mismatches that "disturb" the first command as a second command classic algorithm's model mismatches. This solution imposes designed of classic algorithm with a corresponding robustness reserve. For this reason designing of the second algorithm is made in two steps:

- designing of a classic algorithm base on a model identified in a real functioning point – selected fortuity or, on the middle of process characteristic;
- verification of algorithm's robustness and improving of this, if it is necessary in a new (re)designing procedure;

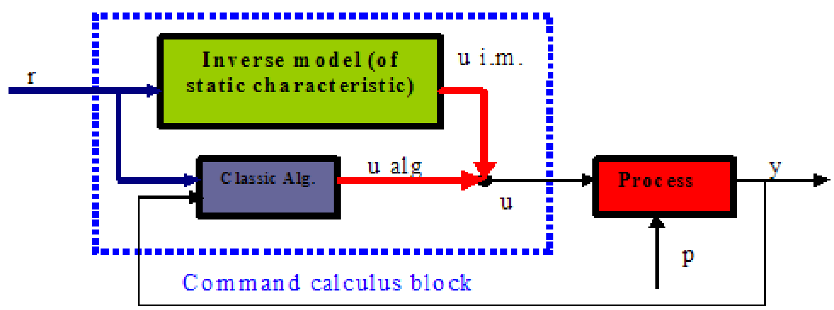

Fig. 1. Proposed scheme for inverse model structure

On Fig. 1, the blocks and variables are as follows:

- Process – physical system to be controlled;
- Command calculus – unit that computes the process control law;

- Classic Alg. – control algorithm (PID, RST);
- y – output of the process;
- u – output of the Command calculus block;
- u alg. – output of the classic algorithm;
- u i.m. – output of the inverse model block;
- r – system's set point or reference trajectory;
- p – disturbances of physical process.

Related to classical control loops, inverse model control need addressing some supplementary specific aspects:

- Determination of static characteristic of the process;
- Construction of inverse model;
- Robust control law design.

On next sections we will focus on the most important aspects meted on designing of the presented structure.

2. Inverse model design procedure

As is mentioned above, for inverse model control proposed structure, the supplementary specific aspects are: determination of static characteristic of the process, construction of inverse model and robust control law design. We will present these on next sections.

2.1. Determination of static characteristic

This operation is based on several experiments of discrete step increasing and decreasing of the command *u(k)* and measuring the corresponding stabilized process output *y(k)*. The command *u(k)* covers all possibilities (0 to 100% in percentage representation). Because the process is disturbed by noises, usually the static characteristics are not identically. The final static characteristic is obtained by meaning of correspondent position of these experiments. Fig. 2 present this operation. The graphic between two "mean" points can be obtained using extrapolation procedure.

According to system identification theory the dispersion of process trajectory can be finding using expression (1):

$$\sigma^2[n] \cong \frac{1}{n-1}\sum_{i=1}^{n} y^2[i],\ \forall \mathrm{n} \in N^* \setminus \{1\} \tag{1}$$

This can express a measure of superposing of noise that action onto process, process's nonlinearity etc. and is very important on control algorithm designed robustness.

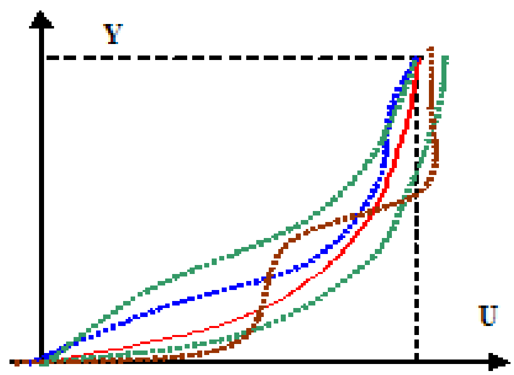

Fig. 2. Determination of static characteristic of the process.
Continuous line represents the final characteristic.

2.2. Construction of inverse model

This step deals with the „transposition” operation of the process’s static characteristic. Fig. 3 presents this construction. According to this, *u(k)* is dependent to *r(k)*. This characteristic is stored in a table; thus we can conclude with this, for the inverse model based controller, selecting a new set point *r(k)* will impose finding in this table the corresponding command *u(k)* that determines a process output *y(k)* close to the reference value.

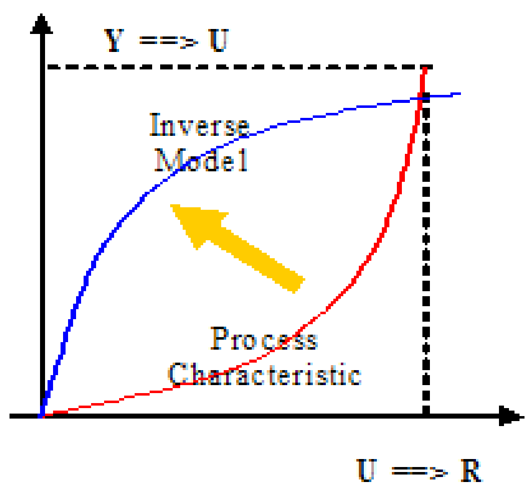

Fig. 3. Construction of inverse model

2.3. Control law design

Control algorithm's duty is to eliminate the disturbance and differences between inverse model computed command and real process behavior. A large variety of control algorithms can be used here, PID, RST, fuzzy etc., but the goal is to have a very simplified one.

For this study we use a RST algorithm. This is designed using pole placement procedure [5]. Fig. 4 present a RST algorithm:

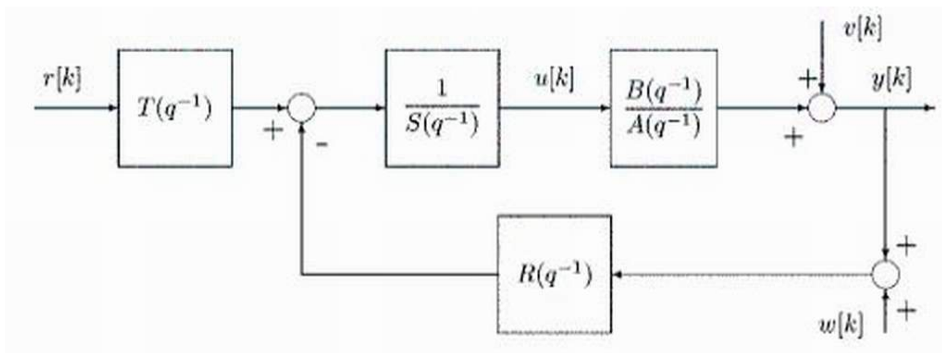

Fig. 4. RST control algorithm structure

Where R, S, T polynoms are:

$$\begin{aligned} R\left(q^{-1}\right) &= r_0 + r_1 q^{-1} + \ldots + r_{nr} q^{-nr} \\ S\left(q^{-1}\right) &= s_0 + s_1 q^{-1} + \ldots + s_{ns} q^{-ns} \\ T\left(q^{-1}\right) &= t_0 + t_1 q^{-1} + \ldots + t_{nt} q^{-nt} \end{aligned} \tag{2}$$

Algorithm pole placement design procedure is based on identified process's model:

$$y(k) = \frac{q^{-d} B(q^{-1})}{A(q^{-1})} u(k) \tag{3}$$

where

$$\begin{aligned} B\left(q^{-1}\right) &= b_1 q^{-1} + b_2 q^{-2} + \ldots + b_{nb} q^{-nb} \\ A\left(q^{-1}\right) &= 1 + a_1 q^{-1} + \ldots + a_{na} q^{-na} \end{aligned} \tag{4}$$

The identification is made in a specific process operating point and can use recursive least square algorithm exemplified in next relations developed in [4]:

$$\hat{\theta}(k+1) = \hat{\theta}(k) + F(k+1)\phi(k)\varepsilon^{0}(k+1), \forall k \in N$$
$$F(k+1) = F(k) - \frac{F(k)\phi(k)\phi^{T}(k)F(k)}{1+\phi^{T}(k)F(k)\phi(k)}, \forall k \in N \tag{5}$$
$$\varepsilon^{0}(k+1) = y(k+1) - \hat{\theta}^{T}(k)\phi(t), \forall k \in N,$$

with the following initial conditions:

$$F(0) = \frac{1}{\delta} I = (GI)I, 0 < \delta < 1 \tag{6}$$

The estimated $\hat{\theta}(k)$ represents the parameters of the polynomial plant model and $\phi^{T}(k)$ represents the measures vector.

This approach allows the users to verify, and if is necessary, to calibrate algorithm's robustness.

Next expression and Fig. 5 present "disturbance-output" sensibility function.

$$S_{vy}(e^{j\omega}) \overset{def}{=} H_{vy}(e^{j\omega}) = \\ = \frac{A(e^{j\omega})S(e^{j\omega})}{A(e^{j\omega})S(e^{j\omega}) + B(e^{j\omega})R(e^{j\omega})}, \quad \forall \omega \in R \tag{7}$$

In the same time, the negative maximum value of sensibility function represents the module margin.

$$\Delta M\big|_{dB} = -\max_{\omega \in R} \left|S_{vy}(e^{j\omega})\right|\Big\|_{dB} \tag{8}$$

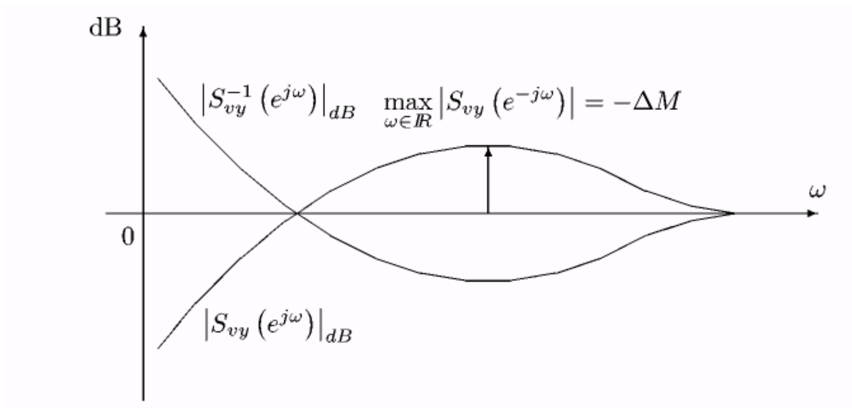

Fig. 5. Sensibility function graphic representation

Base on this value [5], in a "input-output" representation, process nonlinearity can be bounded inside of "conic" sector, presented in Fig. 6, where a_1 and a_2 are calculated using next expression:

$$\frac{1}{1-\Delta M} \geq a_1 \geq a_2 \geq \frac{1}{1+\Delta M} \tag{9}$$

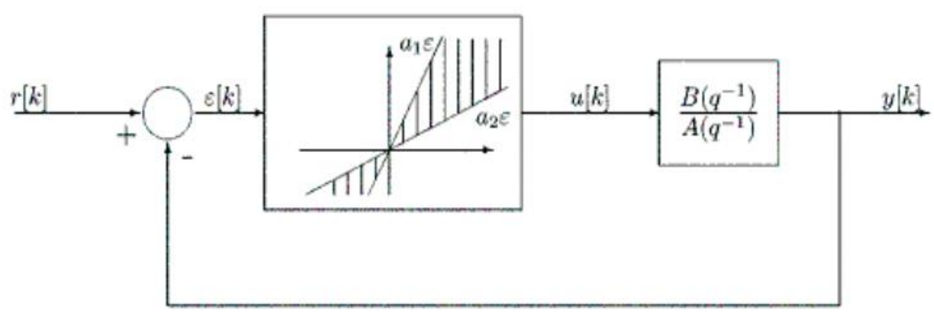

Fig. 6. Robust control design procedure

Finally, if is imposed that all nonlinear characteristics to be (graphicaly) bounded by the two gains, or gain limit to be great or equal to process static characteristic dispersion $\Delta G \leq \sigma$, a controller that has sufficient robustness was designed.

3. Analysis of proposed structure

In this section we will present few advantages, disadvantages or limitation and some possible developing ways of presented structure.

3.1. Advantages of proposed structure

The main advantage consists in using of classics procedure in designing of the control algorithm and determination of inverse command block. There are used well know procedure for identification and control law design. As will be shown in experimental tests all procedures for inverse model characteristic identification can be included in a real time software application.

Because the global command contains a “constant” component generated by an inverse model command block, according to set point value, the system is very stabile. Inverse model command generator can be replaced by a fuzzy logic bloc that can “contain” human experience about some nonlinear processes.

Because the control law is not very complex real time software and hardware implementation don’t need important resources.

3.2. Disadvantages or limitations of the structure

This structure is very difficult to use for the system that doesn’t have a bijective static characteristic and for systems with different functioning regimes.

Another limitation is that this structure can be used only for stabile processes. In situations where the process is “running”, the global command is very possible to not have enough flexibility to control it.

The increased number of experiments for determination of correct static characteristic can be other disadvantages of the structure.

3.3. Possible developing

In situation when the control law becomes very complex, situation cased by difficult determination of process characteristics, the system can be “divided” in two ore more components, becoming a “multiple inverse model system”.

These systems can be easily implemented on PLC structures.

4. Experimental results

We have evaluated the achieved performances of the inverse model control using a designed and implemented process simulator software application, developed on National Instruments’s LabWindows/CVI. On Fig. 7, one can see a simulated positioning control system, its operation medium having variable viscosity. The main goal is to control the piston's position.

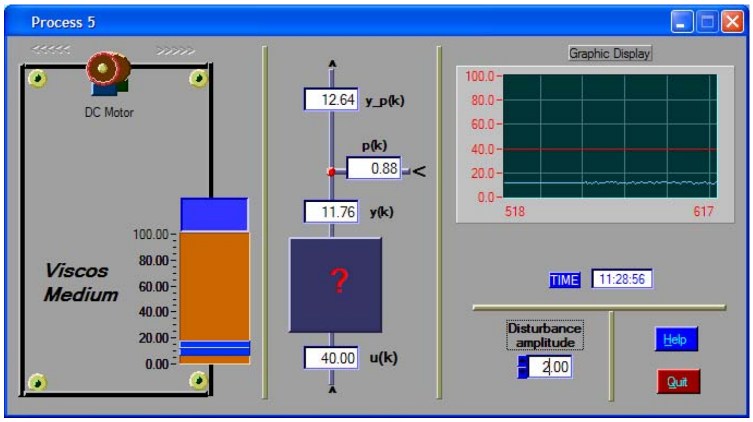

Fig. 7. Process simulator software application

Similar tests were made on an experimental installation presented in Fig. 8, where the position of an object contained in the vertical tube must be controlled using an air flow generator.

Fig. 8. Experimental installation

To verify the inverse model control structure, it was designed and implemented a real-time software application, which can be connected with the process simulator or with experimental installation. This application implement the scheme proposed in Fig. 1.

The user's interface is presented on Fig. 9. Here, on right side, can be observed four slide controls, left to right order: imposed set point value, process value, total command value and "classic" algorithm value. On real time evolution graph it is visible that on stationary regime the "classic" command is very close to zero that means a good determination of inverse model. During transitory regime this command has an increased value necessary to correct the global system's reference tracking performance.

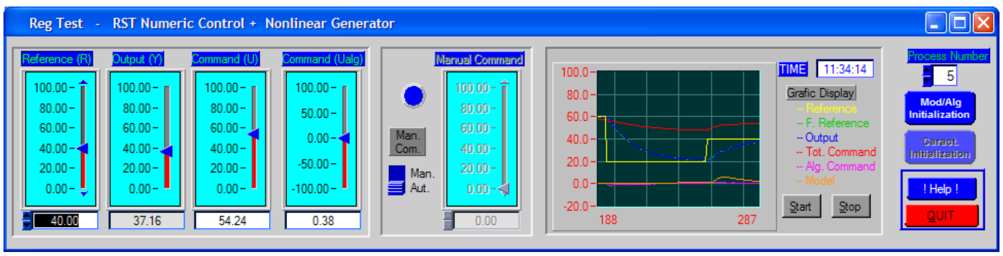

Fig. 9. Inverse model controller real-time software application

In the same time this application allows the user, in a special window, to construct the inverse model (Fig. 10).

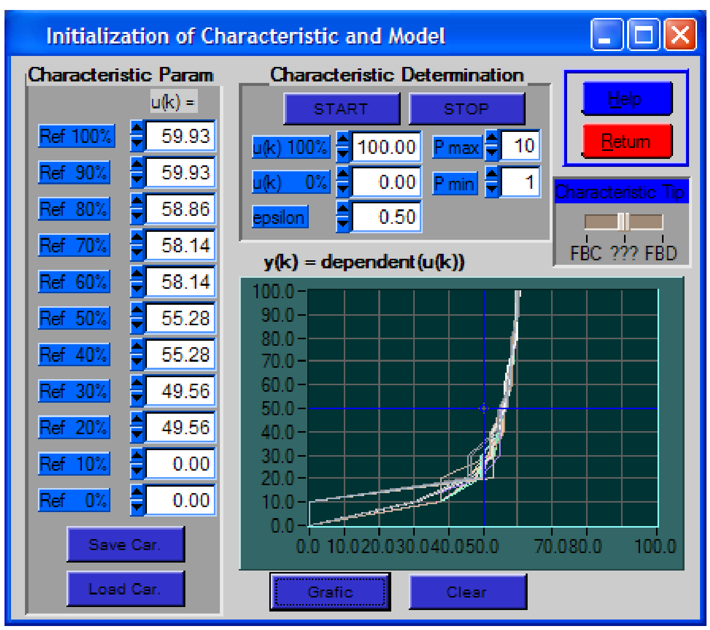

Fig. 10. Inverse-model controller real-time software application – process characteristics determination window

Using this feature there are made 10 tests to determine the process's static characteristics. These are presented in a MS-EXCEL chart (Fig. 11). There are also presented the minimal, medium and maximal characteristics.

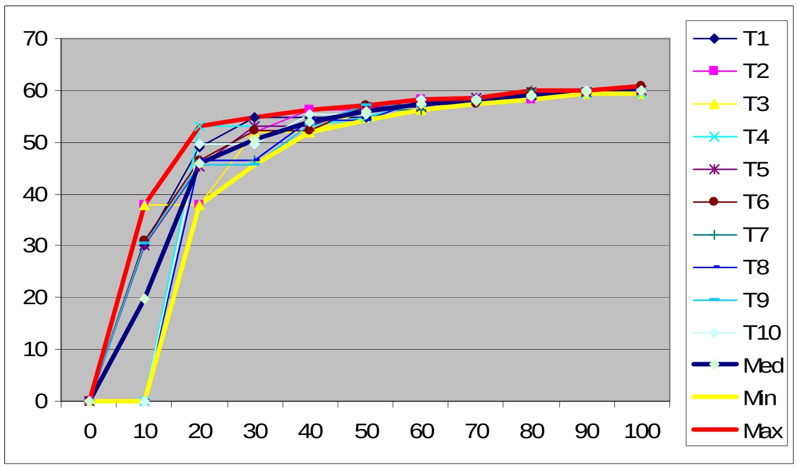

Fig. 11. Inverse model tests representations

For these test the value of dispersion of nonlinear characteristics is 15.81.

Using Te=0.2 s sampling time and Least Square identification method from Adaptech/WinPIM the model is:

$$M = \frac{1.541650}{1 - 0.79091q^{-1}}$$

The corresponding controller (using pole placement):
(for Tracking performances: second order dynamic system with w0=2.0, x=0.95, Disturbance rejection performances: second order dynamic system with w0=1.1, x=0.8, using WinReg)

$$R(q^{-1}) = 0.083200 - 0.056842q^{-1}$$
$$S(q^{-1}) = 1.000000 - 1.000000q^{-1}$$
$$T(q^{-1}) = 0.648656 - 1.078484q^{-1} + 0.456187q^{-2}$$

The close loop performances (using WinReg software application) are presented in Fig. 12:

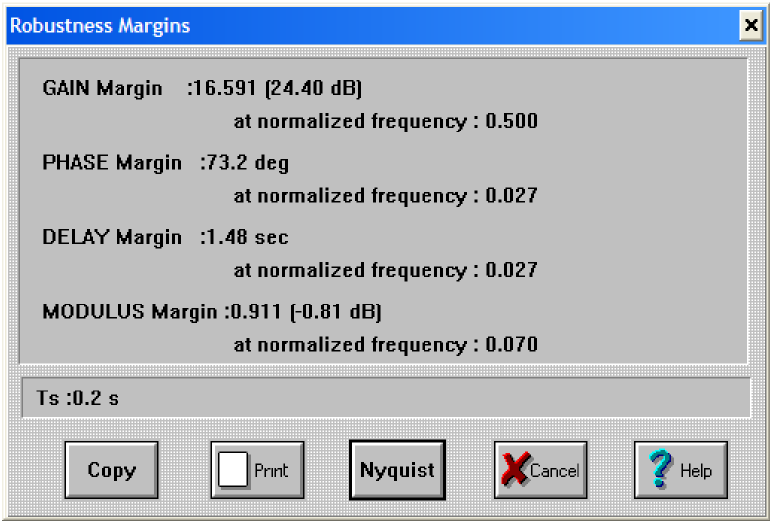

Fig. 12. Close loop performances

It can be observed that gain margin value is 16.591, a greater value that static characteristics dispersion (15.81). For this reason the designed RST algorithm has enough robustness to control the process.

The tests made on real-time functioning prove the structure stability and performances set point changing. Fig. 13 present few of these tests:

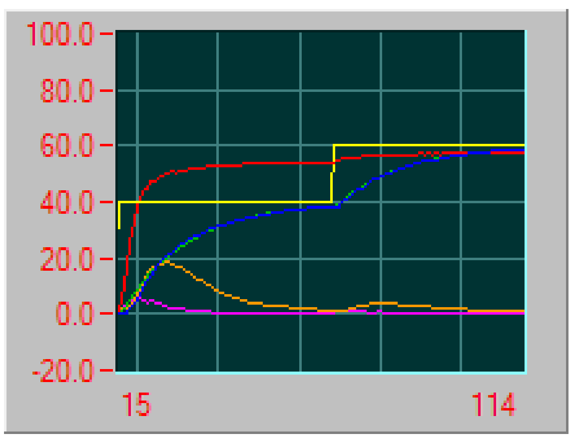

Fig. 13. Testing of structure stability on different functioning point

On this figure the evolution curves are represented using next color code:

- yellow – set point;
- green – filtered set point;
- blue – process output;
- red – control structure output (total command);
- purple – RTS algorithm output;
- orange – identified model output;

In these tests it can be observed that:

- there are no shocks on set point changing;
- the system is stabile on different functioning points;
- after each system stabilization the RST command value decrease to zero that mean that the inverse model is correct determinate;
- Process's output and filtered set point are very close that mean good performances on set point tracking;

5. Conclusions

The paper proposes an inverse model structure as a solution for nonlinear processes. For this structure for each component there are presented the design methods. These are based on original combination of experimental tests and classics identification and close loop pole placement methods.

The performances of the classic algorithm the control law is evaluated using robustness criterions.

There are made some analysis about advantages and disadvantages of proposed structure.

On experimental results section there are presented the evaluated results obtained using a real time software implementation of proposed control structure. The tests are made on a new implemented software simulator and on an experimental laboratory installation.

During exploitation (inverse model) does not impose complex operations, it is very easy to use, but it is limited from the nonlinearity class point of view and processes with variable parameters.

Acknowledgement

This work was supported by IDEI Research Program of Romanian Research, Development and Integration National Plan II, Grant no. 1044/2007.

R E F E R E N C E S

[1] *Dumitrache, I.*, Ingneria Reglarii Automate, Politehnica Press, Bucuresti, ISBN 973-8449-72-3, 2005

[2] *Dussud, M., S. Galichet and L. Foulloy*, Fuzzy supervision for continuous casting mold level control, IFAC, 2000

[3] *Flaus J.M.*, La Regulation Industrielle, Editure Herms, Paris, 1997

[4] *Landau, I.D. and A. Karimi*, Recursive algorithm for identification in closed loop: a unified approach and evaluation, Automatica, vol. 33, no. 8, pp. 1499-1523, 1997

[5] *Landau, I. D., R. Lozano and M. M'Saad*, Adaptive Control, Springer Verlag, London, ISBN 3-540-76187-X, 1997

[6] *Ljung L., Soderstroom T.,* Theory and Practice of Recursive Identification, MIT Press, Cambridge,Massashusetts, 1983.

[7] *Lupu, C., D. Popescu B. Ciubotaru, C. Petrescu, G. Florea,* "Switching solution for multiple-model control systems", 14th Mediterranean Conf on Control Autom., Ancona, Italy, June 28-30, 2006

[8] *Narendra, K. S. and J. Balakrishnan*, Adaptive Control using multiple models, IEEE Transactions on Automatic Control, vol. 42, no. 2, February, page. 171 – 187, 1997

[9] *Pages, O., P. Mouille and B. Caron*, Multi-model control by applying a symbolic fuzzy switches, IFAC, 2000

[10] *Pajunen, G.,* Adaptive control of wiener type nonlinear system, Automatica 28, 781-785, 1992

[11] *Richalet J.*, Practique de la Commande Predictiv, Editura Herms, Paris, 1993

[12] *Tao, G., and P. Kokotovic*, Adaptive control of systems with actuator and sensor nonlinearities, Wiley, N.Y, 1996.